QUEST INTERNATIONAL UNIVERSITY PERAK

FACULTY OF SCIENCES AND TECHNOLOGY

Bachelor of Computer Science (Hons)

BCS3105 CS PROJECT PHASE 1

PROJECT INTERIM REPORT

PREPARED BY:

TAN ZHONG MING (QIUP-201804-002120)

SUBMISSION DATE:

22 MAY 2020

QUEST INTERNATIONAL UNIVERSITY PERAK

DETECT STUTTERED SPEECH BY USING MACHINE LEARNING

by

TAN ZHONG MING

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfilment of the requirements for the degree stated.

Signature: _____

Main Supervisor: Dr. Lee Lam Hong

Signature: _____

Head of Programme: Dr. Sharanjit Kaur

Date: 28/2/2020

DETECT STUTTERED SPEECH BY USING MACHINE LEARNING

By

TAN ZHONG MING

A Thesis

Submitted to the School of Computing

as a Requirement for the Degree of

BACHELOR OF COMPUTER SCIENCE (HONs)

QUEST INTERNATIONAL UNIVERSITY PERAK

IPOH,

PERAK

MAY 2020

DECLARATION OF PROJECT

Title of Project: Detect Stuttered Speech by Using Machine Learning

I TAN ZHONG MING hereby declare that the project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other diploma at QIUP or other institutions.

Witnessed by:

_____ _____

Signature of Author

Signature of Supervisor

Tan Zhong Ming

Permanent address:

Name of Supervisor:

Lot 2285

Mukim Machap, Menggong

78000, Alor Gajah

Dr. Lee Lam Hong

Melaka,

Malaysia

Date: -

Date: -

REVISION HISTORY

Date	Version	Description
14/05/2020	1.0	Initial Version, Chapter 1
16/05/2020	1.1	Chapter 2.1 and 2.2
18/05/2020	1.2	Chapter 3, Methodology, Planning, Use Case Diagram, Use Case Report
19/05/2020	1.3	Chapter 3, Hardware Requirement, Software Requirement, Class Diagram, Activity Diagram
20/05/2020	1.4	Chapter 3 Sequence Diagram, State Diagram, Package Diagram, Deployment Diagram
21/05/2020	1.5	Table of Contents, Acknowledgement, Preliminary Page, Formatting, Appendix, Page numbering
21/05/2020	1.6	Initial commit and push to GitHub
22/05/2020	1.7	Add revision history, push to GitHub
22/05/2020	1.8	Cover page, Declaration of project, Reformat Table of Content, Reformat and rearrange page number, Literature review 2.3 - Speech Recognition and Correction of a Stuttered Speech
23/05/2020	1.9	Remove Speech Recognition and Correction of a Stuttered Speech, add DeepSpeech2: End-to-End Speech Recognition in English and Mandarin
24/05/2020	2.0	Finished Literature Review 2.4, Modify all diagram to match the new design

TABLE OF CONTENTS

ACKNOWLEDGEMENT

First and foremost, I would like to give a heart full of appreciation to my project supervisor, Dr. Lee Lam Hong for his encouragement and continuous support throughout this project as well as his guidance and feedback on reviewed reports.

Secondly, I would like to give another heart-warming thank you to my moderator Dr. Sharanjit Kaur for his views and ideas on the downfalls of my project as well as the need to view my project on different perspectives to get a better understanding.

I would like to thank Ms. Syazwani Yahya, Mr. Cheang Kah Wai, Dr. Noor Hidayah Binti Zakaria and Mr. Tay Yen Pei for always providing a positive attitude and a never-ending support.

I would also like to thank my mentor, Ms. Menaka Sivapalan for all the support, knowledges and words of wisdom leading up to this project and throughout this project as well.

I would also like to thank to my classmate Ms. Low Hwei Li, Ms. Chin Yoke Nien, Mr. Fong Jia Hui, Mr. Ho Jin Heng for sharing their knowledge to me and navigate me through certain complication.

Last but not least, I would like to thank my classmates and course mates for always being there for me and providing a positive vibe.

A million thank you is not enough for me to express the appreciation for all the help I have received.

ABSTRACT

The traditional method to remove a stuttered speech from video required a lot of human effort. The video editor must play the whole video and watch numerous times to remove the stuttered speech. Apart from that, to remove a stuttered speech from the footage required to learn editing tools such as Adobe Premiere, Sony Vegas, Hitfilm and more. The learning curve for editing tools is very high, and they are different from each other due to different concept and different workflow. To remove a stuttered speech, the user must learn the timeline, timecode, audio wave and more. Next, existing speech recognition is not able to detect the stuttered speech, so when the user requests the API, it will return a weird result.

In this project, a camera app will develop and implementing machine learning. The goal of the app is using machine learning to identify and remove the stuttered speech from the video. It can reduce the learning curve significantly and reduce the human effort required to edit a video. First, the audio will extract from the video and undergo pre-processing algorithm such as amplify and normalization. Pre-processing audio can provide a better result for the next step. Next is implement the MFCC algorithm to extract the audio features. After generating the features for each data in the dataset, the features will ingest to a neural network SVM for training purpose. The trained model will save in a specific format file protobuf. Then, the protobuf will include as an asset in the android app. By using the Android SDK and Android NDK, we can use the function provided by TensorFlow Lite to classify the input by using this special format file. The stuttered part in the video will be labelled by a remove algorithm to generate a timestamp. Then the video will undergo an auto subtitle algorithm will generate the subtitle base on the generated timestamp. After that, the app will render the video base on the timestamp and store it in mp4 format with the subtitle.

PRELIMINARY PAGE

List of abbreviations

No.	Term	Description
1.	ANN	Artificial Neural Network
2.	API	Application Programming Interface
3.	ASR	Automatic Speech Recognition
4.	CER	Character Error Rate
5.	CPU	Central Processing Unit
6.	CTC	Connectionist Temporal Classification
7.	DAT	Digital Audio Tape
8.	DS	Deep Speech
9.	DTW	Dynamic Time Wrapping
10	FFT	Fast Fourier Transform
11	Fps	Frame per second
12	GB	GigaByte
13	GMM	Gaussian Mixture Model
14	GPU	Graphic Processing Unit
15	GRU	Gated Recurrent Unit
16	GUI	Graphical User Interface
17	HDD	Hard Disk Drive
18	HMM	Hidden Markov Model
19	HPC	High Performance Compute
20	k-NN	k-nearest neighbors' algorithm
21	LDA	Linear Discriminant Analysis
22	LPCC	Linear Prediction Cepstral Coefficient
23	LSTM	Long-Short Term Memory
24	MFCC	Mel-frequency cepstral coefficient
25	ML	Machine Learning
26	MLP	Multi-Layer Perceptron
27	MP	Megapixel
28	n-D	n-Dimension (1D = 1 Dimension)

29	nm	nanometer
30	NN	Neural Network
31	OS	Operating System
32	Py	Python
33	RAM	Random Access Memory
34	ROM	Read-only Memory
35	SSD	Solid State Drive
36	STT	Speech-To-Text
37	SVM	Support Vector Machine
38	TB	TeraByte
39	TTS	Text-To-Speech
40	TV	Television
41	UCLASS	University College London Achieve of Stuttered Speech
42	UI	User Interface
43	WER	Word Error Rate

List of Tables

List of Figures

List of Equations

1 INTRODUCTION

1.1 Background

1.1.1 Stutter

According to Cambridge dictionary, stuttering is a speech disorder when someone try to speak something with difficulty especially the first part of a word. For instance, pause before the sentence ends or repeating the same word or phonemes several time. Speech is one of the communication methods used by human to express their feelings, idea and thoughts. Stuttering also known as stammering. According to (Vikhyath Narayan K N, S P Meharunnisa, 2016), there is approximate 1 % of population in the world faces stuttering problem. Stuttering is a common speech disorder when the patients in a tense environment. There are several types for the dysfluencies. Repetition is one type of dysfluencies which include syllable repetition, whole world repetition and phrase or sentence repetition. The following type of dysfluencies is prolongation, interjection and pauses. Repetition occur when a syllable or sound is repeated at the beginning of the word. For example, “the baby-baby ate the soup” and “W-W-W- Where are you going?”. Prolongation occur when the speaker prolongs the sounds or syllable such as “The baaaaaaby ate the soup”. Interjection commonly use “um” and uh” to fill up the pause sector also known as filled pause or fillers. For example, “the baby um uh ate the um soup”.

1.1.2 Video

According to (Celie O'Neil-Hart, Howard Blumenstein, 2016), 6 out of 10 people is prefer to watch video online rather than television. According to (Youtube for Press, n.d.) the number of channels owns more than a million subscribers were grown by more than 65%. The number of video views grew 120,589,156 on 21 April 2020 for the highest number of subscriber channel, T-Series. T-Series channel grew around 100k video views daily. The growing of youtube significantly shows the importance of online video and shows the demand for a video editor. Most of the Youtuber does hire at least one or more edit, or to edit their video daily. However, to become an editor required a large amount of skill and years of training such as composition, colour science, graphic design skill, sound processing and various skill. According to Linus Tech Tips channel on YouTube, the average salary for his video editor is $29 per hour, and he has 7 video editors. It does cost $420,000 for a year.

1.1.3 Machine Learning

According to (Tom Mitchell, 1998), machine learning is a well-posed learning problem which mean a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. Machine learning is a new capability for computers such as data mining, self-customizing program, application can't program by hand for example Natural Language Processing. Machine learning divided into two main category which are supervised learning and unsupervised learning. There is more category from that, such as reinforcement learning, recommender systems and more. Supervised learning divided into two categories. First, regression problem is to predict a valued output based on the input such as house price. Supervised learning does provide a right answer at the end compare to unsupervised learning. Second, classification problem is to classify the object base on the parameter given. Common classification problem is to classify have or don't have breast cancer based on the tumor size. The classification problem can become more complex based on the situation. Support vector machine (SVM) is introduced to solve when there is two-group of classification problem. (Stecanella, 2017). Unsupervised learning is one of the machine learning algorithms. The goal of unsupervised learning is to draw inferences base on the datasets given without any labeled responses. In other meaning is ask the machine to classify and provide an inference based on the dataset given. The common algorithm included hierarchical clustering, k-Means

clustering, Gaussian mixture models, Self-organizing maps, Hidden Markov models. (MathWorks, n.d.)

1.2 Problem Statement

1.2.1 The editor must remove the stuttered speech manually by using existing tools

Existing tools are not capable of removing shuttered speech from the video automatically. So, the editor must playback the video to identify the shuttered speech and remove it automatically. Most of the time editor will play the video numerous times to ensure there is not stuttered speech in the video. Remove manually will cause the editor to spend most of the time just for removing the stuttered part instead of the content. The cost to run a media company will also increase due to the long edit time. According to Linus Tech Tips channel, hire a video editor will cost $29 per hour, but most of the time editor just removing the stuttered speech and find out the successful part to make a video.

1.2.2 Inaccurate auto generate subtitle

Current autogenerate subtitle tools will affected by the various variable such as background noise, low quality of mic, accented speech and more. Apart from that, current auto subtitle generator not able to detect the full stop and chunk the longer speech into smaller pieces. This will cause the algorithm to generate long text which are unable fit into a small screen. Next is existing subtitle generator does not provide appropriate timestamp and any pre-processing so it will transfer the responsibility to the app developer to develop a pre-process algorithm to improve the accuracy.

1.3 Objectives

1.3.1 To identify and remove stuttered speech by using machine learning.

The traditional method is to identify the stuttered speech by listening and watching the video numerous times. A trained machine learning model will be capable of identifying the stuttered speech without any human involvement, including the threshold. As a comparison, a machine learning method can identify the stuttered speech in a short amount of time. Another side, human has to listen and watch the video over and over again. The trained machine learning model will identify the stuttered speech and provide a timestamp for remove stuttered speech algorithm. The second benefit after remove the algorithm is to provide a more reliable speech recognition by using API. Existing speech recognition API not able to detect the stuttered speech so this project also provides a tool for better speech recognition by removing the stuttered speech from video. Apart from that, after remove the stuttered speech from the video it can reduce the size of file that required send to API for speech recognition. In other meaning, this will save a lot bandwidth when calling the speech recognition API.

1.3.2 Improve the accuracy of speech recognition

Base on my audio processing experience and testing, denoising the audio and normalize the audio will bring the best benefit for using API which included accented speech. A second machine learning model will implement in this project to remove the background noise from the audio to improve the accuracy of subtitle generation. An improved version of auto subtitle algorithm will be developed which are capable to chunk the audio properly. Deep Speech 2 proposed by Baidu has great potential to deal with accented speech and noisy speech. According to (Amodei et al., 2016), Deep Speech (DS) is an end-to-end deep learning system. The idea is to train the neural network directly from the raw speech by using convolution technique in RNN. Current system is complicated because it is designed deal with phonemes and syllable. Base on their study, the model they proposed is perform close to human performance in speech recognition. Apart from that, the goal of deep speech is to include as many language as they can by using only one engine.

1.4 Scope

1. Perform pre-processing techniques to audio data set for better sound quality
2. Build and train an Artificial Neural Network (ANN) for the stuttered speech classification model.
3. Build and train an End-to-end deep learning model - Deep Speech
4. Develop an application and algorithm for Android to automatically remove the stuttered speech.
5. Implement audio pre-processing technique to improve the accuracy of speech recognition.
6. Develop an improved version of auto subtitle generator.

1.5 System Scope

Learn how to remove stuttered speech and improve auto subtitle generator by using machine learning.

1.6 Benefits and Significance

This project will improve the process for video editing process and subtitle by implementing Machine Learning. This project will develop an android application to automate the manual process in video editing and providing a better auto subtitle generator.

The table below shows the comparison between the current video editing process and proposed video editing app.

	Current Video Editing Process	Proposed Video Editing App
Process	Detect and remove the stuttered speech manually and import one by one from the library	Detect and remove the stuttered speech automatically and in batch by using machine learning
Transcribe	Denoise the speech manually includes normalize, hard-limit, equalizer, noise remove and more.	Denoise and pre-process the speech automatically by using machine learning in batch
Process time	High	Low
Human involvement	High	Low

Table 1 Comparison of Current Video Editing Process and Proposed Video Editing App

The proposed method will significantly improve the process and process time for video editing. Apart from that, the proposed method will improve the accuracy of subtitle generation. Machine learning will introduce to automate and improve video editing process and auto generate subtitle.

1.7 Constraint & Limitation

1.7.1 Mobile Platform

1.7.1.1 Processing power

The machine learning task is hard to run on mobile devices due to its limited processing power. Mobile devices are providing mobile services which allowed user still connected when they move around. Run a machine learning task on the mobile devices will consume a lot of battery life and reduce mobility because the user cannot move around during the execution time.

1.7.1.2 Connectivity

Mobile devices are a handheld device which provides a high mobility computing device for a user. High mobility may cause internet disruption, so it is hard to run a cloud computing on a server through mobile devices due to unstable connectivity of mobile devices.

2 LITERATURE REVIEW

2.1 A Comparative Study of Recognition Technique Used for Development of Automatic Stuttered Speech Dysfluency Recognition System

Author: Swapnil D. Waghmare, Ratnadeep R. Deshmukh, Pukhraj P. Shrishrimal, Vishal B. Waghmare, Ganesh B. Janvale and Babasaheb Sonawane

DOI: 10.17485/ijst/2017/v10i21/106092

According to (Waghmare et al., 2017) , there is many researchers successful detect the stuttered speech by using various technique. In this paper, they summarized the methods, statistical analysis, findings, application and improvement. According to this paper the dysfluency in speech will affect the performance of automatic voice recognition and makes the system usable for user. In this paper, they mentioned few types of disorder. First is language disorder, it's included cluttering disorder which means the speaker have limited knowledge to the particular language and cause rapid cause. But cluttering disorder does not include repetition or hesitation. There is three type of speech disorder which include Aprexia (Dyspraxia), Articulation and Stuttering. Aprexia speech also known oral-motor speech disorder. The patient facing apraxia speech has a problem with muscle movement and cause difficult to turn the speech into words. Articulation disorder occur when a people unable to pronounce speech sound properly and below its mental edge, also known as artic disorder. Stuttering included three type of symptoms which are linguistic, physiological and neurophysiological. This paper mentioned three type of stuttering which are development stuttering, neurogenic stuttering and psychogenic stuttering. In this paper also mentioned about the type of dysfluencies such as interjection, revisions, prolongation and broken words. Apart from that, there is many types of stuttered speech database we can use to train our database. The most commonly used is UCLASS Stuttered Speech Database. It is collected by Psychology Department of University College London (UCL) over last 20 years. UCLASS is released in two different version. Version 1 included 18 females and 120 male speakers from 5 years 4 months to 47 years. UCLASS version only contain monologues recording. For the version 2 contain monologs, reding and spontaneous conversation. But the age ranging is smaller, it is from 5 years 4 months to 20 years 7 months. According to this paper there is different language of stuttered speech database such as Polish Language, Northern Sotho Language, Indian Regional Language (Kannada) and German Language. Artificial Neural

Network (ANN) is inspired by biological counterpart, this can use to solve many complex problems in real-world. ANN is widely used for stuttered speech detection and classification of fluent and dysfluent in stuttered speech. According to this paper, the combination of Support Vector Machine (SVM) and Mel-frequency cepstral coefficients (MFCCs) have the second highest accuracy for detection which is around 98.00% with 16 samples from UCLASS. The classifier that successfully applied to stuttered speech detection include ANNs, HMMs, Hopfield Network, MLP, Korhonen, Perceptron, SVM, k-NN, LDA, DTW and GMM.

For the ANN, this paper mentioned one of the researchers record the speech by using DAT tape and down sampled to 20kHz and transferred to the computer. ANNs gave 78.07% of accuracy for both prolongation and repetition dysfluent speech. Some of the researcher analyzed the speech samples by using FFT 512 with the 21 digital 1/3-octave filters of center frequency between 100Hz to 100kHz.

Support Vector Machine (SVM) is powerful for pattern recognition to classify two classes in higher dimensional space. They proposed a method to detect syllable repetitions automatically. The assessment divided into four stages which are segmentation comparison, feature extraction, score matching and decision logic. The SVM used to identify the normal speech and dysfluent speech. The system can provide 94.35% of accuracy with 15 dysfluent speech sample. 12 samples are used to train the neural network and another 2 is used to evaluate the accuracy of the model. Next researcher trains the k-NN, LDA and SVM model by using UCLASS database, it includes 43 individual speaker and 107 recording. The sounds sample was down sampled to 16000Hz. Three model provided around 95% accuracy.

Hidden Markov Model (HMM) is the most common model used to detect the stuttered speech such as prolongation and repetition. Two paper presented in 2007 shows 70% of accuracy to detect the stuttered speech. In their next paper they got the best result which are 80% of the accuracy in 2010. They sample the sound at 22050Hz with the same dynamic range -50dB

Apart from that, there is two paper published in 2009. They evaluate the accuracy of the k-NN and LDA by using MFCC and LPCC feature extraction method for detecting the stuttered speech. They used 10 recording from the UCLASS database and down sample to 16kHz. They achieved 90.91% by implementing 10^{th} fold cross-validation to MFCC feature in LDA. They split the feature set into 60:40 ratio for training and evaluating the model. They

repeat the experiment 10 times for each k-values. Then they achieve 89.77% of accuracy for k-NN and 87.50% for LDA. Next, they also tested on two different feature extraction method. Based on their study, 25 MFCC features provide the best accuracy 92.55%. LPCC is slightly better than MFCC which are 94.51%.

The formula below is the measurement for severity of stuttered speech:

$$\text{Syllables Per Minute (SPM)} = \frac{\text{Total number of syllables read}}{\text{Total times in seconds}} \times 60$$

Equation 1 Syllables Per Minute

$$\text{Percent Disfluency (PD)} = \frac{\text{Total number of disfluent syllable}}{\text{Total number of syllable}} \times 60$$

Equation 2 Percent Disfluency

The author of this paper claimed, the percentage of the male faces disfluent speech is higher than the female. So, age and gender ratio are one of the important factors to diagnosis the disfluent speech.

The purpose of this paper is to review and collect the type of stuttered speech detection system, method of processing stuttered speech and type of stuttered speech database. In this paper, they discussed various stuttered speech database that developed in different language. Apart from that, they found the stuttered speech database is commonly used for analysis and recognition purpose.

2.2 Detection and Analysis of Stuttered Speech

Author: Vikhyath Narayan K N, S P Meharunnisa

Link: http://ijarece.org/wp-content/uploads/2016/04/IJARECE-VOL-5-ISSUE-4-952-955.pdf

Stuttering also known as stammering. (Mecs, 2016)It is a speech disorder. According to this paper, there is 1% of population in the word faces stuttering problem and the female ratio is four times higher than male. This paper introduced a new method to classify the stuttered speech such as incomplete phrases, repetition, prolongation, interjection, silent pause and broken words by using MFCC feature extraction and SVM. The author achieves 90% accuracy for dysfluent speech and 96.67% for fluent speech. There are five steps involve to classify the stuttered speech proposed by the author.

First is the input speech. Author in this paper use UCLASS database as the input speech. The UCLASS recording is save in .wav format. The author analyzes and classify the input speech in this step.

Second is signal pre-processing. Pre-emphasis is a popular and tricky signal pre-processing tool due to the computing limitation in those days. Pre-emphasis is reducing the amplitude of lower frequency band and increasing the amplitude of higher frequency band. Pre-emphasis will provide a slightly better result rather than apply nothing. According to author, the pre-emphasis will help to deal with DC offset that presented in recordings to improve the voice activity detection. But current speech recognition system does not require pre-emphasis due to the current algorithm. For example, cepstral mean normalization at the final stage should not have any effect.

Third is syllable segmentation. Syllable segmentation is the ability to identify the syllable of words or phrases. It's included identify the component of words or phrases by using auditory, visual or numerical presentation. For example, one-syllable "bee-zzz" can helps children to understand the concept of syllable. Author proposed a new algorithm which will automatically split the continuous speech signal into multiple syllable segments. The algorithm implements a process based on a positive function – short-term energy function. Hence, the short-term energy function is a positive function, so it is similar to the magnitude of spectrum. So, algorithm able to determine the segment boundaries by using group delay

processing of the magnitude spectrum. Author claimed that, this is an essential steep in speech recognition.

The figure below shows the flowchart for segmentation:

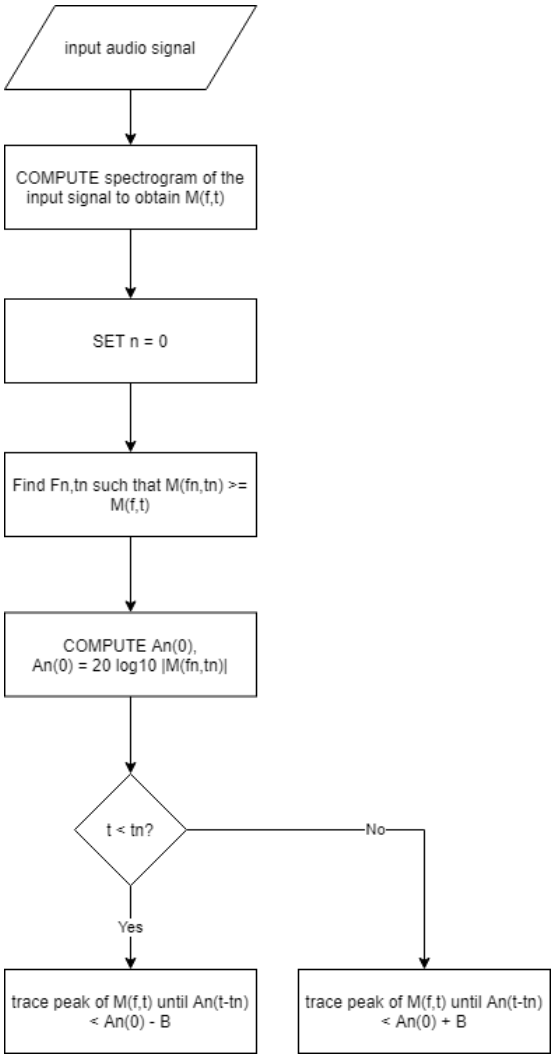

Figure 2.1 Flowchart for Segmentation, Adapted from "Detection and Analysis of Stuttered Speech", by Vikhyath Narayan and S P Meharunnisa, April 2016

Fourth step is feature extraction. The author recommended MFCC feature extraction method. The traditional method in automatic speech recognition is using 10 – 12 coefficients to coding the speech. Although MFCC is sensitive to noise but it can be solved by utilizing the information in the recurrence of speech signals. For the frequency below 1kHz, MFCC will use non-linear frequency scale.

The figure below shows the MFCC block diagram:

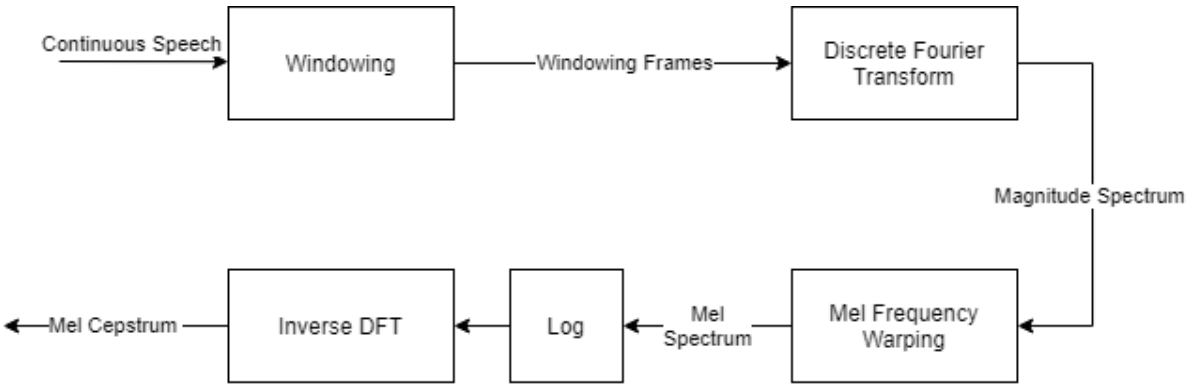

Figure 2.2MFCC Block Diagram. Adapted from "Detection and Analysis of Stuttered Speech", by Vikhyath Narayan K N and S P Meharunnisa, April 2016

SVM is a supervised machine learning model and commonly used for recognize the pattern, classification problem and regression analysis. The basic SVM contain multiple input neuron and output two possible classification. SVM model will represent the examples as point in a space then mapped into different category with a clear gap. Kernel trick a technique that map the multiple inputs into a high-dimensional feature space implicitly. Kernel trick allow model perform non-linear classification efficiently. But the most common kernel is linear kernel, polynomial kernel and gaussian kernel.

In this paper, they proposed a reliable indicator for abnormal speech. For example, to classify the stuttered speech by using MFCC feature and SVM machine learning model. They successful classify repetition, prolongation and interjection with 96.67% accuracy.

2.3 Speech Recognition and Correction of a Stuttered Speech

Author: Ankit Dash, Nikhil Subramani, Tejas Manjunath, Vishruti Yaragarala and Shikha Tripathi

DOI: 10.1109/ICACCI.2018.8554455

According to (Dash et al., 2018), stuttering is an voluntary action will affect the fluency of speech. It includes repetition of syllable, involuntary prolongation and involuntary pauses. The purpose of this paper is to proposed a detailed method to improve existing stuttered speech detection system. Author claimed most of the paper briefly report the method to detect stuttered speech without more detail information. In this paper also include the solution for issues and method to detect and correct stuttered speech in an acceptable time. Author proposed a few solutions to identify different type of stuttered speech. First is prolongation, author used amplitude threshold to eliminate prolongation in stuttered speech. Second, author proposed existing Text-to-Speech (TTS) system to remove the repetition from samples. The method mentioned above will provided a better result for speech recognition.

The goal of this paper is to develop an integrated system that able to identify and correct the stuttered speech. This system is aimed to help for whose facing stuttering problem to control their device by using speech. The method included two algorithms. First algorithm is to remove prolongation by reduce the amplitude to zero when the amplitude of samples is lower than the threshold. Second algorithm is used to remove the repetition of word or syllable by using speech to text technique. The algorithm will remove the repeated word and convert back to speech. These two algorithms are the key to remove stuttered speech in a framework.

There are several techniques for feature extraction, classification techniques, models and databases. Author claimed Chee et al highlight the strength and the weakness of numerous technique that used to detect stuttered speech. Next, Surya and Varghese proposed a supervised learning model which capable to detect and correct the stuttered speech by using text-to-speech techniques. They convert N audio signals into an audio array. Then extract the feature from samples by using MFCC technique. After that, they segmented the audio sample word by word from the dataset manually. Then they use the extracted MFCC feature to train the SVM model since the speech recognition is multiclass problem. They gain 76% of accuracy for the testing. The limitation of the system is only able to predict trained words.

Apart from that, a simplest method is proposed by another researcher. The researcher trains the neural network by using maximum amplitude as the input and threshold amplitude to determine stuttered speech as the output. The advantages of this method are the system able to remove the prolongation and silence pause and reconstruct the speech. But according to author, this method only achieves 62% of accuracy. The downside of this method is it will remove the non-stuttered parts from the samples.

The last method to detect the stuttered speech that review by (Dash et al., 2018) is convert the stuttered speech into equivalent text. Then convert back to normal speech by using artificial neural network. This method can eliminate the silence pauses from samples with 80% of accuracy. Apart from that, this method also capable to recognize a compete sentences instead of trained words. But the limitation of this system is not able to remove the repetitions of the speech.

There is limited work has been done for stuttered speech. The traditional method required massive computational power to train the neural network. It is not practical and feasible to apply the method that mentioned in above. Apart from that, HMM and MFCC is less accurate due to its need number of observations for HMM to predict the stuttered part.

So, author proposed a new method which is efficient to remove the silent pause and prolongation by using amplitude. Then, correct the speech by using few strings operations. Author develop and testing the proposed method by using MATLAB and GUI offers the user to either remove only prolongations or other types of stuttered speech. It’s included repetition, prolongations and interjections.

The figure bellow shows the proposed algorithm:

1. Obtain the speech sample with a duration of six seconds from the user, either through recording or as an audio clip.
2. Through functions from MATLAB, obtain the maximum amplitude of the samples.
3. Pass the maximum amplitude value to the created single neural network to obtain an appropriate threshold. The threshold obtained is always a function of the maximum amplitude.
4. Divide the speech sample into 300 short overlapping frames of equal length.
5. Discard the frames of the signal with amplitude less than the threshold value.
6. Merge the retained frames. Write the corrected audio to a new file sample.
7. After prolongations in the speech sample are removed, convert the new file sample to equivalent text, and write the text to a new text file using Java. This process is done to remove any existing repetitions of words or phrases
8. Perform a check on the file, at word level, for any undesired repetitions of words or phrases or characters.
9. Remove all the detected repetitions from thestring.
10. Write the corrected text to a new text file and convert the new text to speech using any of the existing TTS systems. Write the corrected speech to a new file sample.

Figure 2.3 Proposed Stutter-free system algorithm

The figure below shows the design methodology of stuttered speech processing system:

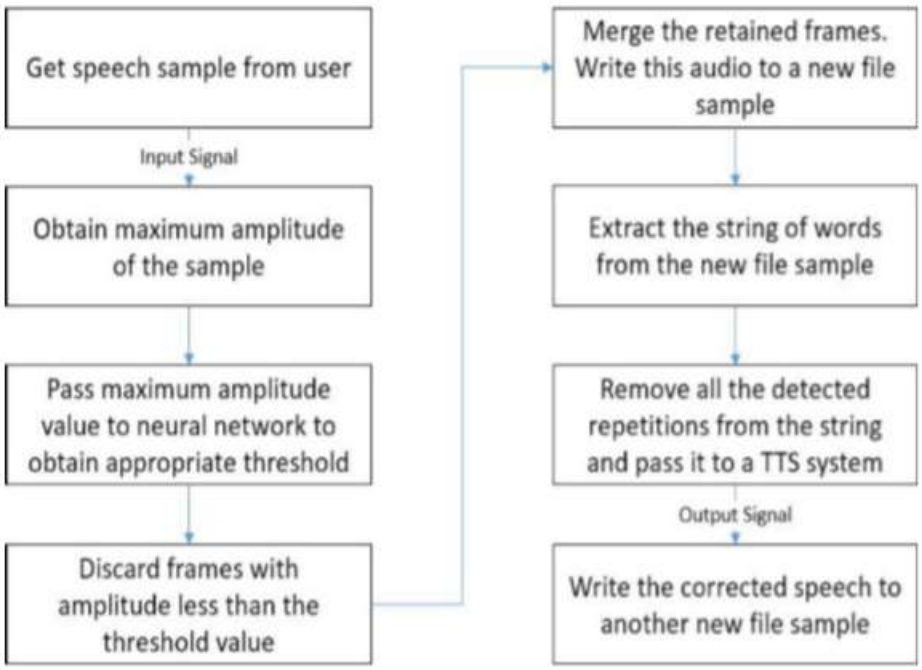

Figure 2.4 Design Methodology of the Stuttered Speech Processing System

They use 60 samples out of 110 to train the neural network. Each sample is 6 seconds. Back propagation algorithm is used in this neural network to generate a proper amplitude threshold for each trained sample. The amplitude threshold can remove the prolongation efficiently but not for repetition. So, string repetition removal algorithm will take places to remove the repetition words up to phrase-level. 50 different speech samples are used to evaluate the system. They successfully obtain 43 correct output out of 50 samples.

The table below show the performance of the stuttered speech processing system:

Number of tested samples	Correct output Obtained	Incorrect output obtained
50	43	7

Table 2 Performance of the developed stuttered speech processing system

The accuracy obtained is 86%. 5 to 8 seconds is required to obtain an amplitude threshold when processed a 6 seconds speech with 8000Hz sampling rate and 300ms frames by using 2.6GHz Dual Core I7 machine. The trained sentence can reduce the processing time within 5 seconds.

In conclusion, the proposed system successfully obtained 86% of accuracy. Two algorithms are implemented to increase the efficiency and accuracy of the system to remove stuttered speech.

2.4 Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin

Author: (Amodei et al., 2016)

arXiv: 1412.5567

Introduction

A deep learning model which capable to recognize either Mandarin or English is proposed in this paper. This is an end-to-end deep learning model. This model is aimed to replace the existing speech recognition system and deal with accents, noisy environment and different language problem. This system implements High Performance Compute (HPC) technique and successful speed up 7 times compare to previous system developed by them. Apart from that, this system is designed to deploy in server and able to deliver result within short amount of time. So, Batch Dispatch with GPU technique is used in their data center.

This paper said modern Automatic Speech Recognition (ASR) is overcomplicated and only focus on one particular language. It will make speech recognition process become more complicated, time consuming and less flexible. Human has the robustness to deal with accented speech and noisy environment, without additional training. An ASR is made up of multiple component such as complex feature extraction process, acoustic models, language model, pronunciation model, speaker adaptation and more. But these components will make the tuning become difficult and hard to implement a different language model. In other mean, the models are not generalized and adapt to other language like human. They believe they can make a one single engine which able to deal with all the languages from scratch without dramatic changes.

The goal of this paper it to developed an end-to-end deep learning system which able to handle all the problem mentioned in above. So, two very different language is selected in this experiment which are Mandarin and English. In this paper, they tried to improve the performance of previous work by focusing in three areas. The area including model architectures, large labelled training datasets and computational. They train the neural network to predict speech transcription from audio with Connectionist Temporal

Classification (CTC) loss function. Multilayer recurrent connections, convolutional filt0ers and non-linearities is consider apply to RNN especially Batch Normalization (BatchNorm).

They train the deep learning system by using 9,400 hours of Mandarin speech and 11,490 hours for English. 3-6 weeks required to train a single model by using 10 of exaFLOPs cluster. Apart from that, they use 8 or 16 GPUs to achieve 10 of exFLOPs. They improve the efficiency for training the data set by improving the scalability. They implemented CTC loss function on GPU and a custom memory allocator to speed up the training process. Finally, the overall of the system achieve 50 teraFLOP/s when using 16 GPUs. The total of training time required to train the dataset is reduced to 3 to 5 days.

The goal of this system is to reach a human-level performance but not only for specific benchmark. The author claimed this system is perform better than human in some commonly-studied benchmarks. Batch dispatch allowed the system improve the efficient which closed to real-time performance for their Mandarin Engine. The compute latency achieves 67ms at 98th percentile and 10 samples are loaded to server simultaneously.

Preliminaries

Symbol	Meaning
$x^{(i)}$	Input - sequence
$y^{(i)}$	Output – corresponding label
$T^{(i)}$	Time-series length
t	Time
l	layer
c	Windows size
i	i-th activation
h	Hidden layer

Table 3 Preliminaries symbol table 1

The meaning of $x^{(i)}$ is the input for the neural network in this case x represent spectrogram, (i) represent the index for the matrix. $y^{(i)}$ represent the corresponding output (label) for $x^{(i)}$. For additional information, the $^{(i)}$ doesn't represent any mathematical operation such as power, it just an index to indicate the position of data in a dataset. So, the training set will represent in the form of $X = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots$ *"Each utterance* $x^{(i)}$ *is a time-series of length of* $T^{(i)}$ *where every time-slices is a vector of audio features,*

$x_t^{(i)}, t - 0, \ldots T^{(i)} - 1$ " The feature they used is a spectrogram of power normalized audio clips, so $x_{t,p}^{(i)}$ represent the power of the p'-th frequency bin in the audio frame at time t. The goal of the RNN is convert the $x^{(i)}$ input sequence into corresponding $y^{(i)}$ label to achieve learning effect.

The equation show below is the function of convolutional layer $h^l = \overrightarrow{h^l} + \overleftarrow{h^l}$:

$$h_{t,i}^l = f(w_i^l \circ h_{t-c:t+c}^{l-1})$$

Equation 3 Convolutional layer function

The l represent the n-th layer of hidden layer. So, h^0 is the first layer of hidden layer, it's represented the input x.

The equation show below is the activation function for convolutional layers:

$$\overrightarrow{h_t^l} = g(h_t^{l-1}, \overrightarrow{h_{t-1}^l})$$

Equation 4 Forward in time recurrent layer activation

$$\overleftarrow{h_t^l} = g(h_t^{l-1}, \overleftarrow{h_{t+1}^l})$$

Equation 5 Backward in time recurrent layer activation

The $\overrightarrow{h^l}$ represent the activation function of forward in time and $\overleftarrow{h}^l$ represent the backward activation function in time. The convolutional layers include one or mode bi-directional recurrent layer.

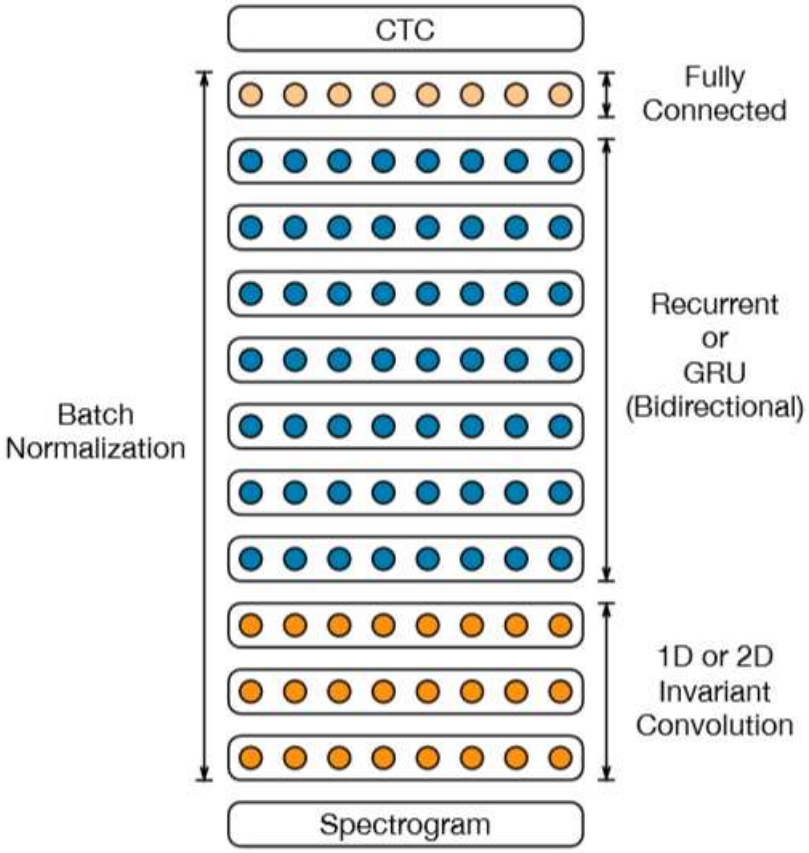

Figure 2.5 Neural Network Diagram of Deep Speech 2

Deep Speech 2 contain total of 11 layers. 3 layers is 1D or 2D invariant convolution layers. Following by 7 layers of recurrent or GRU bi-directional and uni-directional layers. The last layer is fully connected layer to perform prediction based on previous layer.

The output activation for the layer $h^l = \overrightarrow{h^l} + \overleftarrow{h^l}$ is sum of two sets:

$$\overrightarrow{h_t^l} = f(W_t^{l^{-1}} + \overrightarrow{U^l}\,\overrightarrow{h_{t-1}^l} + b^l)$$

Equation 6 Sum of 2 sets of activation

Symbol	Meaning
W^l	Input-hidden weight matrix
$\overrightarrow{U^l}$	Recurrent weight matrix
b^l	Bias term

Table 4 Preliminaries symbol table 2

The functions $g(\cdot)$ can be the standard recurrent operation. It also can use to represent more complex recurrent units such as LSTM and GRU. The input-hidden weight is shared for both direction of the recurrence in this case.

The following equation is the output layer L which is a softmax computing a probability distribution over characters given by

$$p(\ell_t = k|x) = \frac{\exp(w_k^L \cdot h_t^{L-1})}{\sum_j \exp(w_j^L \cdot h_t^{L-1})}$$

Equation 7 Calculate probability distribution at output layer L

They use the CTC loss function to train the model by given an input-output pair (x, y).

Batch Normalization for Deep RNNs

According to their study, the Batch Normalization did increase the performance of deep neural network rather than increase the total number of consecutive bidirectional recurrent layers. They insert a BatchNorm into the feed-forward layer by using $f(\beta(Wh))$ instead of $f(Wh + b)$, where

$$\mathcal{B}(x) = \gamma \frac{x - \mathrm{E}[x]}{(\mathrm{Var}[x] + \epsilon)^{1/2}} + \beta.$$

Equation 8 BatchNorm at feed-forward layer

They also consider another method which is insert the BatchNorm before every non-linearity. So, based on the equation 6, the equation will become:

$$\overrightarrow{h}_t^l = f(\mathcal{B}(W^l h_t^{l-1} + \overrightarrow{U}^l \overrightarrow{h}_{t-1}^l)).$$

Equation 9 Implement BatchNorm at Equation 3

This equation makes the mean and variance will accumulated over a single time-step for each minibatch. Author claimed this technique will not increase the performance so they consider another equation which will accumulating an average over successive time-steps. But this equation does make the neural network become more complicated and increase the time complexity.

Author said the BatchNorm is a good approach to train the neural network, but it did not suitable to implement for ASR system.

Frequency Convolutions

Frequency convolutions is introduced 25 years ago. The temporal convolution technique is widely use to improve the efficiency of ASR. Author claimed the convolution in frequency and time-domain can slightly improve the performance of ASR. So, they try to add frequency convolutions between layer 1 and layer 3. This generated 35 million parameters due to the convolution layers add small fraction of parameters to the network. 2D invariance include time and frequency domain. In another side, 1D invariance include only time-domain. The 2D invariance convolution layer does improve the Word Error Rate (WER) by 23.9% for the noisy dataset.

	Dev no LM		Dev LM	
Stride	Unigrams	Bigrams	Unigrams	Bigrams
2	14.93	14.56	9.52	9.66
3	15.01	15.60	9.65	10.06
4	18.86	14.84	11.92	9.93

Table 5 Comparison of WER with different amounts of striding for unigram, bigram on a model with 1 layer of 1D-invarient convolution, 7 recurrent layers and 1 fully connected layer

Striding

They applied longer stride and wider context in the neural network to speed up the training time by reducing the number time-step required. Down sampling the input speech does reduce the time for training but it will also reduce the performance of speech recognition. Striding can implement directly to the Mandarin models but not English. Because the English models will reduce the performance if the striding employ directly. In order to solve this problem, they enrich the English alphabet by using symbols because the output layer of the neural network require at least one time-step per output character. So, they use non-overlapping n-grams because it is easy to construct. Apart from that, non-overlapping bigrams does reduce the length of the output transcription and reduce the length of unrolled RNN. They also transformed the unigram labels into bigram labels by using isomorphism technique. So, the sentence for “the cat sat” will segmented into [th, e, space, ca, t, space, sa, t] when employ non-overlapping bigrams. The last character became unigram

and space will treated as unigram also. This isomorphism structure can ensure the same words are always form by the same bigram and unigram. Apart from that, in Table 5 shows the result for bigram and unigram systems for different level of striding. They observe the larger level or striding does not have significant impact on Word Error Rate (WER). This technique is able to significantly reduce the time-complexity and the memory usage.

Row convolution and Unidirectional Models

They choose uni-directional models instead of bidirectional RNN. Because bidirectional RNN is hard to implement in online with low-latency settings. This limitation is caused by the bidirectional RNN is operate on an entire sample. So, forward-only RNN layers is introduced in the system. To achieve the unidirectional models in speech recognition, a special layer is required row convolution. It only required small amount of future information to make the prediction without any performance lose. This layer is placed above all the recurrent layers. This layer brings the benefit of finer granularity, better Character Error Rate (CER) and good feature presentations.

Data

They train the neural network by using 95 minutes speech per epoch for English and 25 minutes for Mandarin. This technique reduced 10-20% of total training time. Their training data is collected from Wall Street Journal (WSJ), Switchboard, Fisher, LibriSpeech, Baidu Read and Baidu Mixed. So, they have total 11,940 hours of speech for English. We can obtain the data from Linguistic Data Consortium for WSJ, Switchboard and Fisher dataset. Baidu dataset is only for internal use and LibriSpeech is the only free dataset in this paper.

Dataset	Speech Type	Hours
WSJ	read	80
Switchboard	conversational	300
Fisher	conversational	2000
LibriSpeech	read	960
Baidu	read	5000
Baidu	mixed	3600
Total		11940

Table 6 Summary of the datasets used to train DS2 in English.

For Mandarin system, they used 9,400 hours of labelled audio to train the data. But Mandarin data is only for Baidu internal use. Total they gain 8 million utterances for English and 11 million utterance for Mandarin labelled audio.

Some of the dataset for English and Mandarin cause some problem due to the datasets were created from raw data. Some of the clips has the length from several minutes to more than hour. So, they have to unroll the dataset in RNN. They developed an alignment, segmentation and filtering pipeline which will generate a training set that have shorter utterance. This pipeline will align the audio with the transcription and remove the long silence audio to reduce the total length of utterance. At the end of this pipeline, the erroneous example that caused by from a failed alignment will remove from the dataset. This pipeline successful reduced the WER from 17% to 5% and retain more than 50% of the samples in the same time.

To add the robustness of the system, they added the noise in to the existing dataset to generate more data. The technique increases the number of data by manipulating the existing data called Data Augmentation. They find the best ratio to add the noise of the utterance is 40%. The noise source will be randomly selected from a database, then combined with the original audio clip. Each dataset is trained up to 20 epochs and WER decreased by 40%.

The table below show the comparison of WER for two speech system and human level performance on read speech:

Read Speech			
Test set	DS1	DS2	Human
WSJ eval'92	4.94	3.60	5.03
WSJ eval'93	6.94	4.98	8.08
LibriSpeech test-clean	7.89	5.33	5.83
LibriSpeech test-other	21.74	13.25	12.69

Table 7 Comparison of WER for two speech systems and human level performance on read speech.

The table below comparing WER of the DS1 system to the DS2 system on accented speech:

Accented Speech			
Test set	DS1	DS2	Human
VoxForge American-Canadian	15.01	7.55	4.85
VoxForge Commonwealth	28.46	13.56	8.15
VoxForge European	31.20	17.55	12.76
VoxForge Indian	45.35	22.44	22.15

Table 8 Comparing WER of the DS1 system to the DS2 system on accented speech.

The Table 8 shows the system has the performance close to human performance. DS2 does improve for all the accented speech compare to DS1. DS2 can improve further by training by using more accented speech.

The table below show the comparison of DS1 and DS2 system on noisy speech:

Noisy Speech			
Test set	DS1	DS2	Human
CHiME eval clean	6.30	3.34	3.46
CHiME eval real	67.94	21.79	11.84
CHiME eval sim	80.27	45.05	31.33

Table 9 Comparison of DS1 and DS2 system on noisy speech.

They use CHiME datasets to test the noisy speech performance. This dataset includes many noisy environments such as bus, café, street and more. CHiME does include 1320 utterance from WSJ test set. DS2 does improve when compare to DS1, but it performs worse than human in noisy environment. This phenomenon maybe caused by the dataset is using synthetic noise rather than real environment noise.

In conclusion, the End-to-End deep learning has great potential to improve the current speech recognition system. It can improve from numerous way such as increase the data and computational efficiency. The proposed method is highly generic, because the new method can quickly apply to new languages without any significant changes. Numerous neural network and numerical optimization technique is explored in this paper such as SortaGrad, Batch Normalization, larger strides and more. This method able to train the model for full-scale models within few days. They believe this experiment confirm the value of implementing end-to-end deep learning in speech recognition. They hope these techniques will continue growing and make this system outperform than human in most the scenarios.

3 RESEARCH TECHNOLOGY

3.1 Methodology

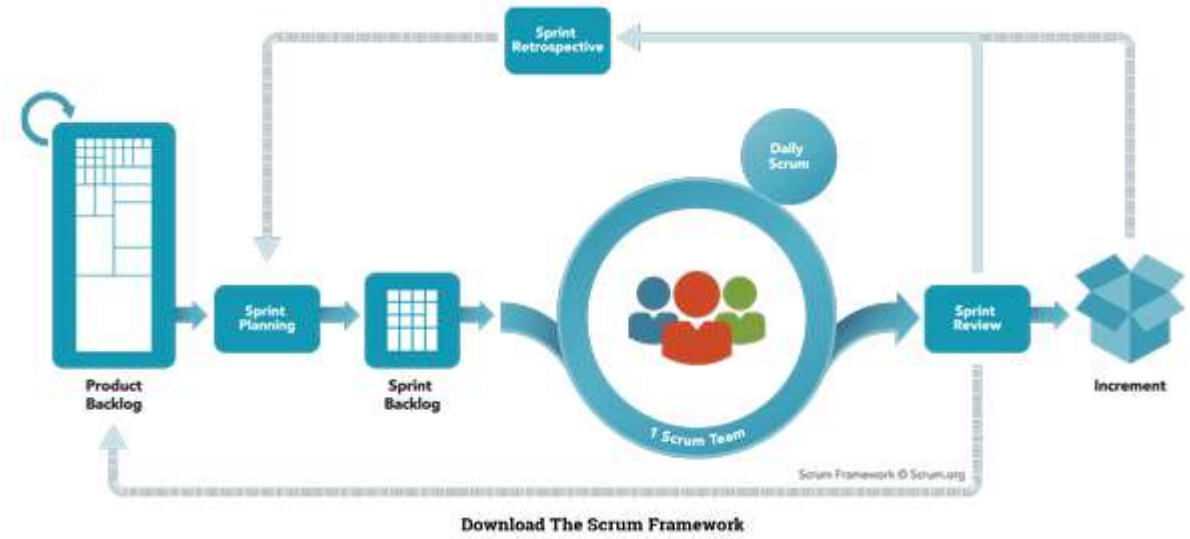

Figure 3.1 - Scrum Model. Adapted from "Nutcache",

retrieved from https://www.nutcache.com/blog/leverage-scrum-to-manage-your-projects/

A scrum model (agile) is implemented in this project. Agile is an iterative approach for software development. The software is developed and delivered to customers in increments. Agile has the flexibility to accommodate frequent changes in the design. Scrum is one of the agile process frameworks which include product owner, scrum master, and development team. Scrum breaks the task into goals that can be completed within the timeboxed iteration, which call sprint. This is a lightweight. Iterative and incremental approach. The sprints should not longer than one month. The development team is self-organized, and responsible convert the backlog into an actual system. Eight members of the development team are required in this project. Product owner representing stakeholders and the voice of the customer. Only 1 product owner is required in this project to maximizing the value delivered by the development team. Scrum Master is responsible for ensuring the Scrum framework is followed and acts as a buffer between the team and any distracting influences. Each team required a scrum master, in this project, there is two development team which are Team A, and Team B. Scrum also included sprint planning, daily scrum, sprint review, sprint retrospective, backlog refinement, cancelling a sprint. These will be implemented in this project. (Fernandes, 2015) (scrum.org, n.d.)

3.1.1 Organization Structure, Role and Responsibility

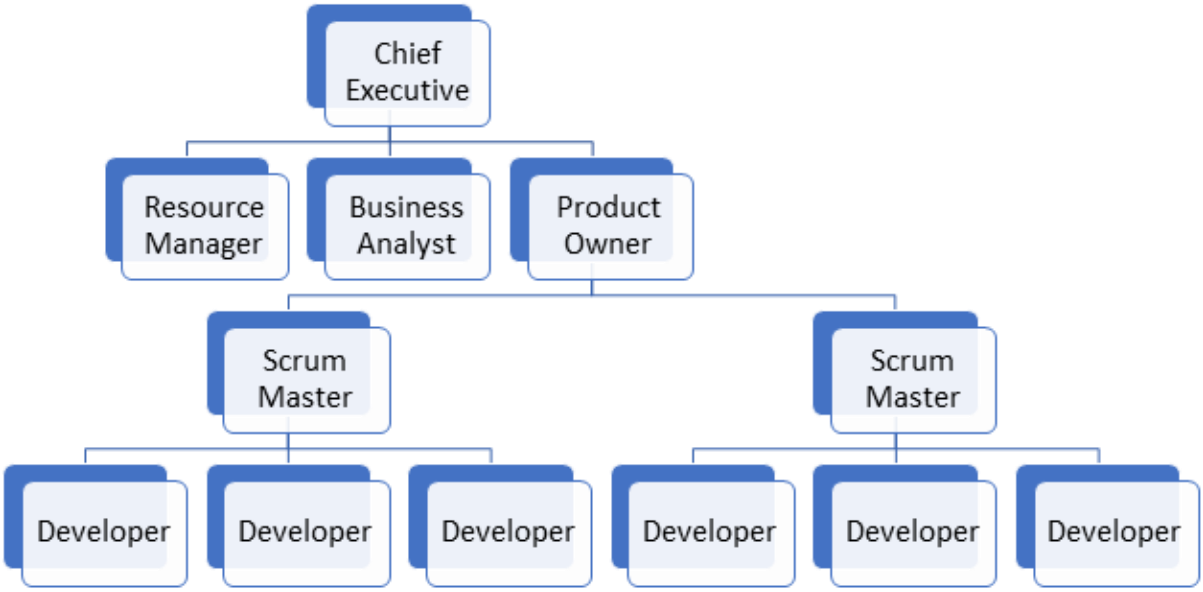

Figure 3.2 Organization Structure for Scrum

Role	Responsibilities
Chief Executive	Manage the management board and communicating with shareholders, government entities and customer, Make
Resource Manager	Manage the resources efficiently and maximize productivity such as manage staff, administer payroll and etc.
Business Analyst	Analysing the business requires flexibility, responsiveness, creativity, innovative thinking, acceptance of change and a dependence on individuals and interactions.
Product Owner	Manages the Product Backlog optimizes the value of the product
Scrum Master	Manages the scrum process and remove impediments
Developer	Self-organize team even there is no Scrum Master and turn the Product Backlog into increments of potentially releasable functionality

Table 10 Role and Responsibilities in Scrum

3.1.2 Definition

Scrum events are defined as the following:

1. **Sprint:** A time-boxed mini project which less than 4 weeks, at the end for each sprint should deliver a releasable product or feature, Sprints include the planning, design, development and testing phase. Each sprint can assign to a synchronized team.
2. **Sprint planning:** A planning stage before the sprint began. Each backlog is prioritizing and review the requirement (product backlog) and created an order list for a particular sprint. Analyze the feasibility for each requirement and features finish at a particular time.
3. **Daily Scrum:** A meeting that held every day morning and takes less than 5 minutes. The scrum master will coordinate the team and discuss their daily goal and achievement. The obstacle also will be discussed in the meeting to seek help from another team. An unclear goal can make the team focus on their daily tasks and increase productivity.
4. **Sprint review:** An informal meeting establishes at the end of the sprint. The increment (product backlog) will be demonstrating to the end-user if any improvement or changes will execute in the next sprint.
5. **Sprint retrospective:** A formal meeting that gathers all the scrum and reviews the sprint. Each sprint will be review in this stage, which included the factor that makes the sprint or goal fail, way to improve the sprint, and etc. Then continue next sprint.

Scrum artifacts are defined as the following:

1. **Story:** Describe what users need to solve their problems. It describes the functionality and the features of the system which is also known as user stories. For example, login, pay and update profile.

Figure 3.3 User Story. Adapted from "Mountain Goat Software", by Mike Cohn, retrieved from http://www.mountaingoatsoftware.com/blog/job-stories-offer-a-viable-alternative-to-user-stories/feed

2. **Product Backlog**: An ordered list of the requirement for the product or features of the system. Product backlog includes all the requirements of the systems or features such as users can pay via credit card. A product backlog is never complete because product backlog will evolve throughout the entire development process. A product backlog is dynamic and frequently changes to fulfil the requirement of the product and what the product needed to be competitive. Adding detail, estimate, and order to items in the product backlog call product backlog refinement. (Fernandes, 2015)
3. **Tasks:** A decomposed of a product backlog. Task refine the product backlog and the requirement of the product or features of the system.

3.2 Planning

3.2.1 WBS

Refer to Appendix A

3.2.2 Gantt Chart

Refer to Appendix B

3.2.3 Deliverables

Deliverables	**Phase**	**Date**
Proposal	Analysis Phase (Planning)	14/2/2020
Documentation (Chapter 1 to 3)	Analysis Phase (Planning – Define Scope)	22/5/2020
Half-working prototype (Audio processing module) Final presentation	Sprint 1	10/6/2020
Refined documentation ANN model Stuttered speech detection model	Sprint 2	25/6/2020
Refined documentation Wavenet model Denoising model	Sprint 3	14/7/2020
Final documentation Final prototype	Sprint 4	18/8/2020

3.2.4 Tools for development

a. Draw.io

Draw.io is free online diagram tools which allow the user to make flowcharts, process diagram, journey map, UML, network diagram and more.

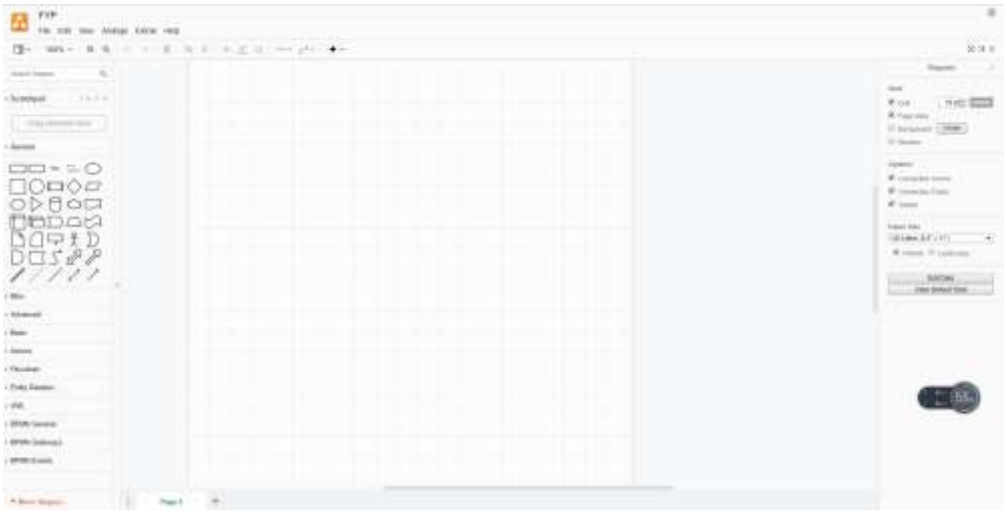

Figure 3.4 draw.io screenshot

b. Anaconda3

Anaconda is a widespread free and open-source distribution of Python and R programming for data science, machine learning, and more. It supports Windows, Linux and macOS. It is easy to use and more reliable and more comfortable to use compared to other distribution. It does support VS Code, JupyterLab, Jupyter Notebook, Powershell Prompt, Qt Console, Spyder, RStudio and a bunch of application and environment.

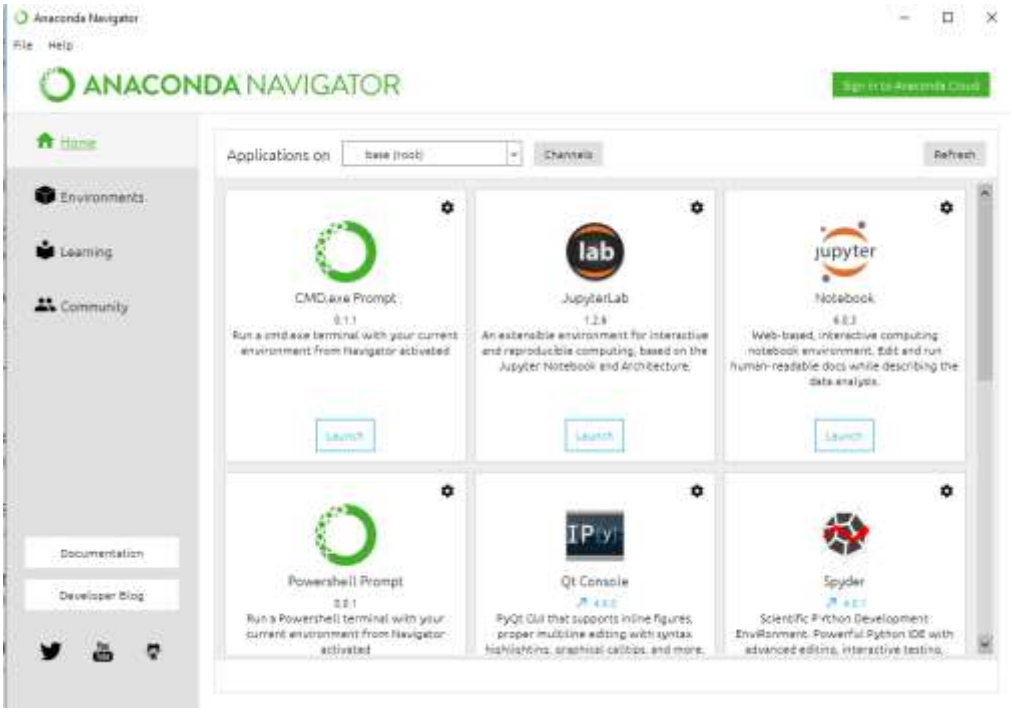

Figure 3.5 Anaconda3 screenshot

c. Jupyter Notebook / Colab

Jupyter Notebook is an opensource web application which allows user to run live code, equations, visualizations and more. It supports a lot of programming language such as Python, Java, R, Octave and more.

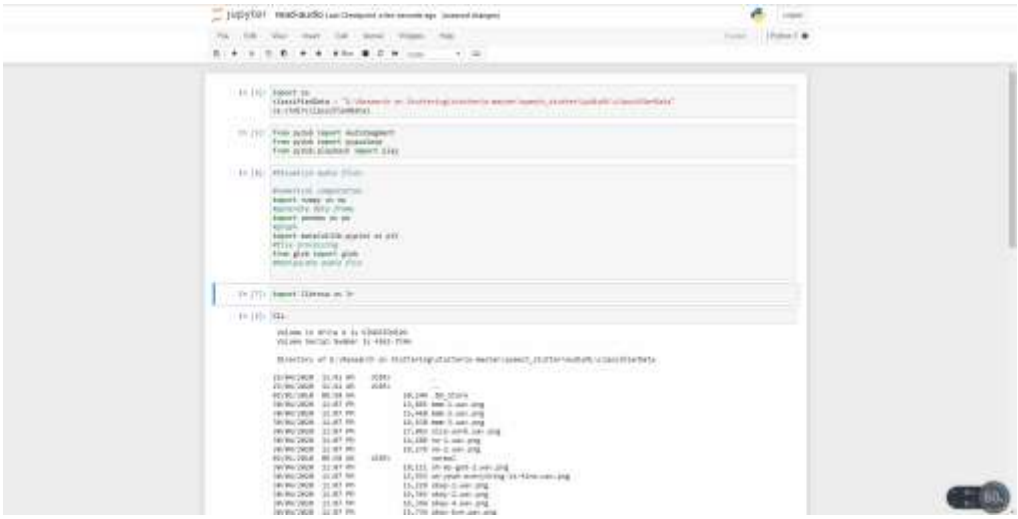

Figure 3.6 Jupyter Notebook screenshot

Colab is a cloud environment provided by Google which allows user to write and execute their python code in the browser. It provided free access to GPUs and

allowed collaborative with other teammates. Colab implements Jupyter Notebook as its user interface.

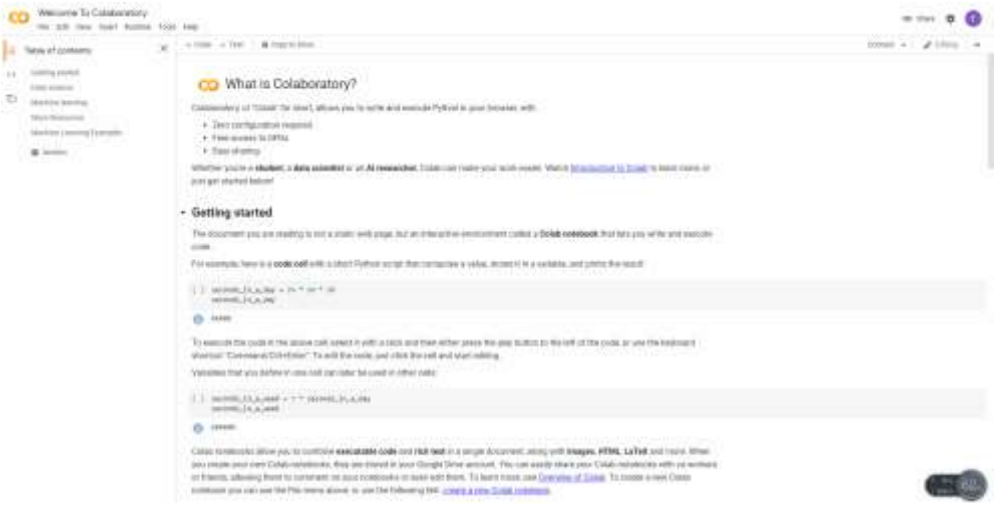

Figure 3.7 Colab screenshot

d. TensorFlow / TensorFlow lite

TensorFlow is an open-source platform commonly used in machine learning and deep learning tasks. TensorFlow is developed by Google Brain team to perform the heavy numerical task. TensorFlow provides Python and C++ API interface for the user for execution. TensorFlow uses C and C++ programming language for backend processing to provide faster processing time. It is based on data flow graphs.

e. Flutter

Flutter is a Google's UI kit which will compile the application for mobile, web and desktop natively. Flutter is a learn once use anywhere concept. Flutter solved the issues when the developer has to launch an application in two different platforms such as Android and iOS. The traditional method to launch an application in two different platform developer must learn the two programming languages which supported by two different platforms. Apart from that, Flutter also solved the issues caused by different screen size for different mobile devices such as iPad, Android smartphone, iPhone, a screen on refrigerator, touch wall, TV Box and more.

3.3 Hardware Requirement

3.3.1 Minimum Requirement

Hardware	Specification
Computer	RAM: 4GB minimum, 8GB recommended Hard disk: 4GB of available disk space Resolution: 1280 x 800 minimum OS: Windows 7 (64bit) or later
Smartphone	ROM: 1GB available RAM: 2GB OS: Android 6.0 (Marshmallow)

Table 11 Minimum Hardware Requirement

The minimum requirement for TensorFlow Lite is Android 6.0. According to (Mobile & Tablet Android Version Market Share Worldwide, 2020), there is only 8.72% people using android 6.0 and 37.4% of people using Android 9.0. So, the test platform will start with Android 6.0 since that is the minimum requirement.

3.3.2 Development and Deployment Environment

Hardware	Specification
Laptop	Model: Acer E15 575G-55Z3 OS: Windows 10 RAM: 8GB GPU: Nvidia GeForce 940MX Processor: Intel i5-7200u Addon: SSD 480GB HDD 1TB
Smartphone	Model: Oppo A57 (CPH1701) Display: 720 x 1280 pixels Platform: Android 6 (Marshmallow) OS: ColorOS3

	Chipset: Qualcomm MSM8940 Snapdragon 435 (28 nm) CPU: Octa-core 1.4 GHz Cortex-A53 GPU: Adreno 505 Main camera: 13 MP, f/2.2, PDAF, 1080p@30fps Internal Storage: 32GB 3GB RAM

Table 12 Development and Deployment Environment

3.4 Software Requirement

3.4.1 Functional Requirement

3.4.1.1.1 System should able to record the video

3.4.1.1.2 System should able to save the recorded video

3.4.1.1.3 System should able to import the video

3.4.1.1.4 System should able to detect and remove the stuttered speech

3.4.1.1.5 System should able to detect and remove the background noise

3.4.1.1.6 System should able to edit the segment of video

3.4.1.1.7 System should able to edit the subtitle

3.4.1.1.8 System should able to preview the video with subtitle

3.4.1.1.9 System should able to render and save the video

3.4.1.1.10 System should able to redo

3.4.2 Non-functional Requirement

Performance	1. System shall not take more than 3 minutes to reboot 2. System shall not require large memory space 3. System shall able to transcribe at least 5-minute video
Usability	1. System shall able to undo 3 times 2. Training time for user shall take less than 30 minutes 3. Each function should have brief description 4. System shall display video in segmented form 5. System shall display subtitle in segmented form
Reliability	1. Mean time to failure of system must more than 1 crash / 10000 use 2. Mean time to recover for system should less than 3 minutes
Operational	1. System must operate in Android platform 2. System must interact with touch screen

Table 13 Non-functional Requirement

3.4.3 User requirement

Video	1. User need delete to remove unnecessary segment 2. User need trim to cut the video 3. User need auto edit to automatically segment the video
Subtitle	1. User need edit to update the subtitle include duration and text 2. User need segmented subtitle so it is easier to use 3. User need remove to remove unwanted subtitle
System	1. User need undo to recover human mistake 2. User need save to save the video to particular folder 3. User need render to save the video with subtitle

Table 14 User Requirement

3.4.4 Minimum Requirement

Android (target platform)	• Android SDK API equal or more then 23 • Android 6.0 or above
Operating System	• Windows 7 or above • Ubuntu 16.04 or above
Driver	• Nvidia GPU drivers CUDA 10.1 requires 418.x or above
Framework	• Python 3.5 or above

Table 15 Minimum software requirement

3.5 SOFTWARE DESIGN

3.5.1 Framework

3.5.1.1 High System work flow

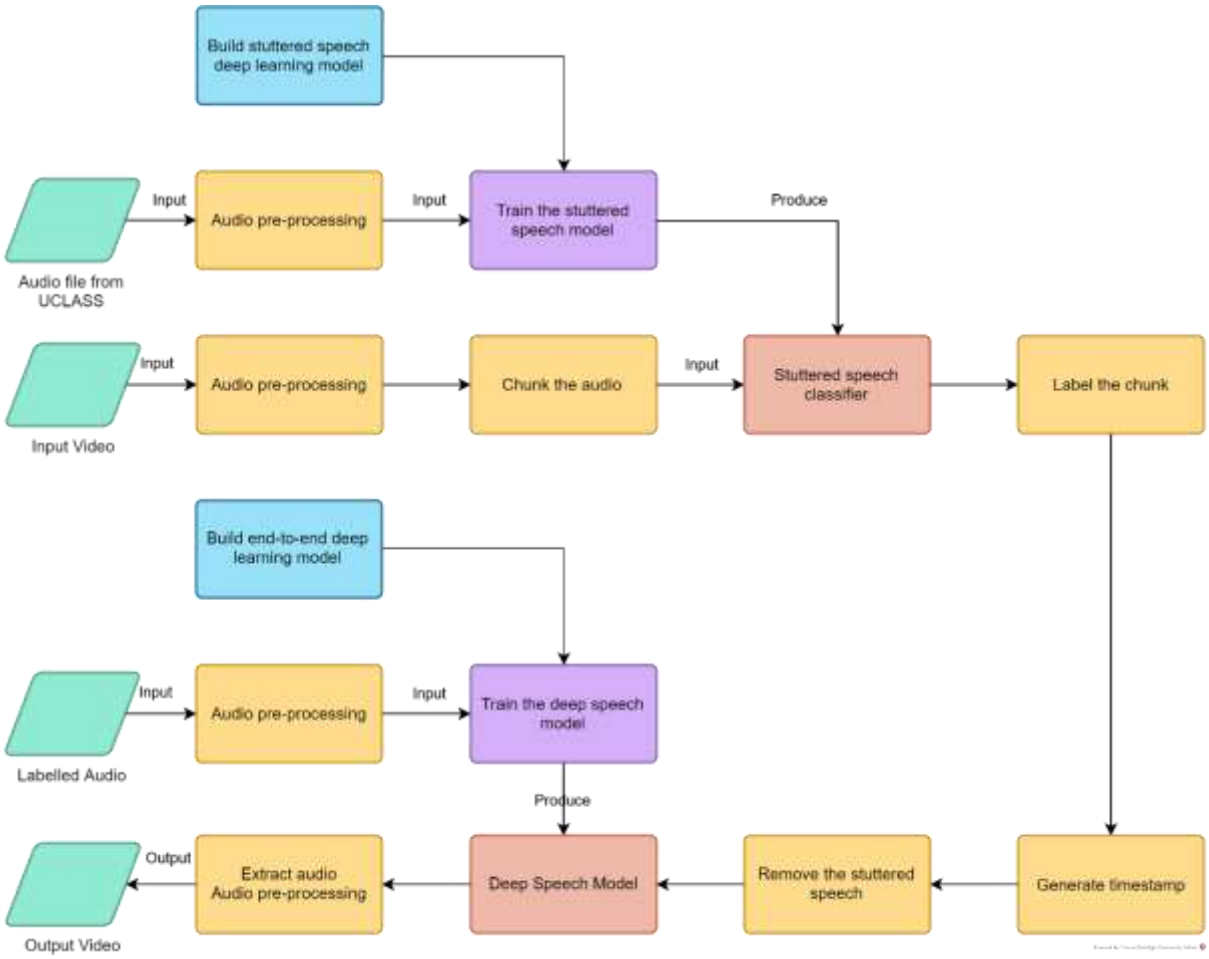

Figure 3.8 High System Workflow

3.5.1.2 TensorFlow Lite

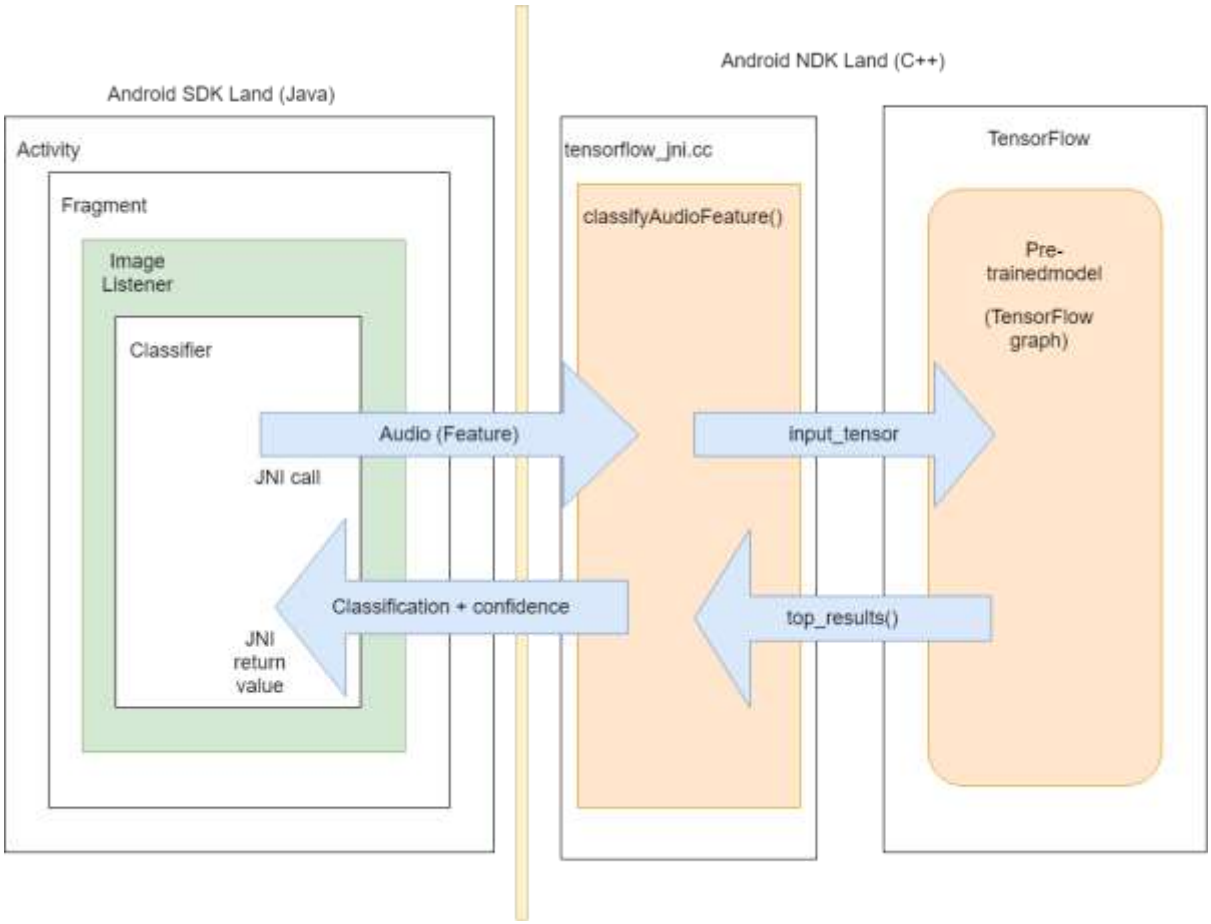

Figure 3.9 TensorFlow lite Framework

Android SDK stands for Android Software Development Kit. Android NDK stands for Android Native Development Kit, which written in C++. Step 1 is the classifier send the audio features to the tensorflow_jni.cc class. tensorflow_jni.cc is a wrapper written in C++ for Android and convert the input to a tensor and resized it. The converted tensor will send to TensorFlow pre-trained model, which call protocol buffer (.pf). Then TensorFlow will return a prediction which is a tensor to the tensorflow_jni.cc file. Then tensorflow_jni.cc will return a list of probability values in an array to Android SDK in Java.

3.5.2 Architecture

3.5.2.1 Use Case Diagram

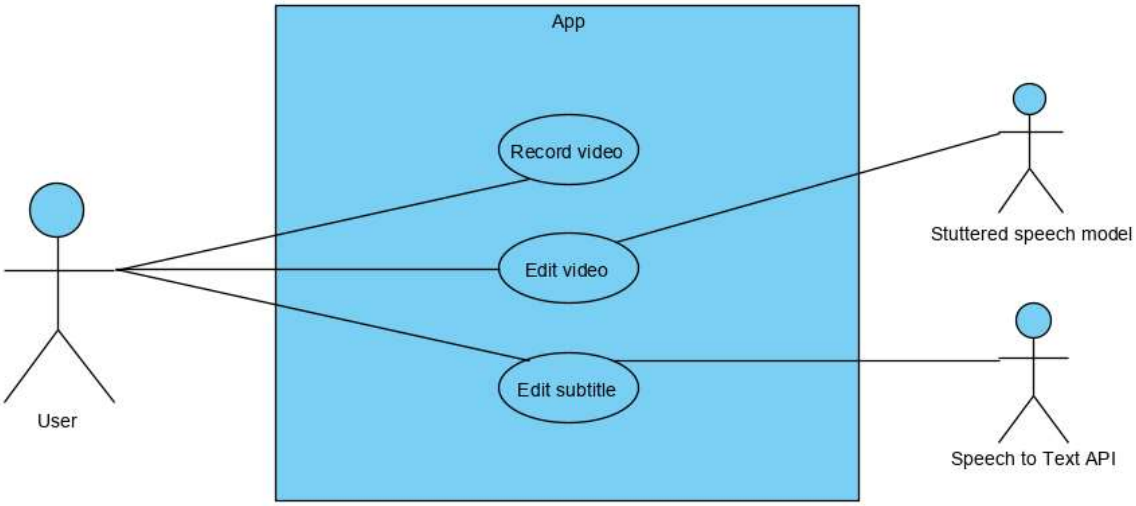

Figure 3.10 Use Case Diagram

3.5.2.2 Use Case Report

Use Case ID	U01.
Use Case Name	Record video
Description	User choose to record a video
Pre-condition	None
Post-condition	Display Menu
Basic Flow	B1. Select Camera [A1] [A3] **[SRS_REQ_01_01]** B2. System process B3. Select Play [A2] [A3] **[SRS_REQ_01_02]** B4. System process B5. Display Menu **[SRS_REQ_01_03]** B6. User case end
Alternative Flow	A1. Select import video [A3] **[SRS_REQ_01_04]** 1. System process [E1] 2. User case end A2. Select delete video [A3] **[SRS_REQ_01_05]** 1. System process 2. Proceed B1 3. User case end A3. Cancel **[SRS_REQ_01_06]** 1. Press <cancel> button **[SRS_REQ_01_07]** 2. System process 3. User case end
Exceptional Flow	E1. Invalid video 1. Display error message **[SRS_REQ_01_08]** 2. Proceed B1 3. User case end

Use Case ID	U02.
Use Case Name	Edit video
Description	User choose to edit a video
Pre-condition	A valid video
Post-condition	Display Menu
Basic Flow	B1. Select delete segment of video[A1] [A3] **[SRS_REQ_02_01]** B2. System process B3. Display menu **[SRS_REQ_02_02]** B4. Select Done [A2] [A3] **[SRS_REQ_02_03]** B5. System process B6. Display menu **[SRS_REQ_02_04]** B7. User case end
Alternative Flow	A1. Select Trim [A3] **[SRS_REQ_02_05]** 1. Enter start time and end time **[SRS_REQ_02_06]** 2. System process [E1] 3. Proceed B1 4. User case end A2. Select Edit **[SRS_REQ_02_07]** 1. Proceed B1 2. User case end A3. Select Cancel **[SRS_REQ_02_08]** 1. Press <cancel> button **[SRS_REQ_02_09]** 2. System process 3. User case end A4. Select Auto Edit **[SRS_REQ_02_10]** 1. Press <auto edit> button **[SRS_REQ_02_11]** 2. System process 3. Proceed B1 4. User case end
Exceptional Flow	E1. Invalid time 1. Display error message **[SRS_REQ_02_12]** 2. User case end

Use Case ID	U03.
Use Case Name	Edit Subtitle
Description	User choose to edit subtitle
Pre-condition	A valid video and a subtitle file
Post-condition	Display Menu
Basic Flow	B1. Display menu **[SRS_REQ_03_01]** B2. Select Done [A1] [A3] **[SRS_REQ_03_02]** B3. System process B4. Display menu **[SRS_REQ_03_03]** B5. User case end
Alternative Flow	A1. Select Segment [A3] **[SRS_REQ_03_04]** 1. Enter start time and end time **[SRS_REQ_03_05]** 2. System process [E1] 3. Enter subtitle **[SRS_REQ_03_06]** 4. System process [E2] 5. Display menu **[SRS_REQ_03_07]** 6. Select OK [A2] **[SRS_REQ_03_08]** 7. Proceed B1 8. User case end A2. Select Edit **[SRS_REQ_03_09]** 1. Press <edit> button [A3] **[SRS_REQ_03_10]** 2. Proceed A1.1 3. User case end A3. Select Auto Subtitle **[SRS_REQ_03_11]** 1. Press <auto subtitle> button **[SRS_REQ_03_12]** 2. System process 3. Proceed B1 4. User case end A4. Select Cancel **[SRS_REQ_03_13]** 1. Press <cancel> button **[SRS_REQ_03_14]** 2. System process 3. User case end

Exceptional Flow	E1. Invalid time 1. Display error message **[SRS_REQ_03_15]** 2. User case end E2. Invalid subtitle 1. Display error message **[SRS_REQ_03_16]** 2. User case end

3.5.2.3 Requirement Traceability List

No.	Requirement ID	Description
1.	SRS_REQ_01_01	Select camera to record
2.	SRS_REQ_01_02	Select play to play the video
3.	SRS_REQ_01_03	Display menu (preview of video)
4.	SRS_REQ_01_04	Select import video from file
5.	SRS_REQ_01_05	Select delete the video
6.	SRS_REQ_01_06	Select cancel
7.	SRS_REQ_01_07	Press <cancel> button
8.	SRS_REQ_01_08	Display error message
9.	SRS_REQ_02_01	Select delete segment of video
10.	SRS_REQ_02_02	Display menu (preview)
11.	SRS_REQ_02_03	Select done
12.	SRS_REQ_02_04	Display menu (preview and share)
13.	SRS_REQ_02_05	Select trim to trim the video
14.	SRS_REQ_02_06	Enter start time and end time
15.	SRS_REQ_02_07	Select edit
16.	SRS_REQ_02_08	Select cancel
17.	SRS_REQ_02_09	Press <cancel> button
18.	SRS_REQ_02_10	Select Auto Edit
19.	SRS_REQ_02_11	Press <auto edit> button
20.	SRS_REQ_02_12	Display error message

21.	SRS_REQ_03_01	Display menu
22.	SRS_REQ_03_02	Select done
23.	SRS_REQ_03_03	Display menu
24.	SRS_REQ_03_04	Select segment of subtitle
25.	SRS_REQ_03_05	Enter start time and end time for segment of subtitle
26.	SRS_REQ_03_06	Enter subtitle
27.	SRS_REQ_03_07	Display menu
28.	SRS_REQ_03_08	Select ok
29.	SRS_REQ_03_09	Select edit the subtitle
30.	SRS_REQ_03_10	Press <edit> button
31.	SRS_REQ_03_11	Select Auto Subtitle
32.	SRS_REQ_03_12	Press <auto edit> button
33.	SRS_REQ_03_13	Select cancel
34.	SRS_REQ_03_14	Press <cancel> button
35.	SRS_REQ_03_15	Display error message
36.	SRS_REQ_03_16	Display error message

Table 16 Requirement Traceability List

3.5.2.4 Flow Chart

3.5.2.5 Activity Diagram

Traditional Method

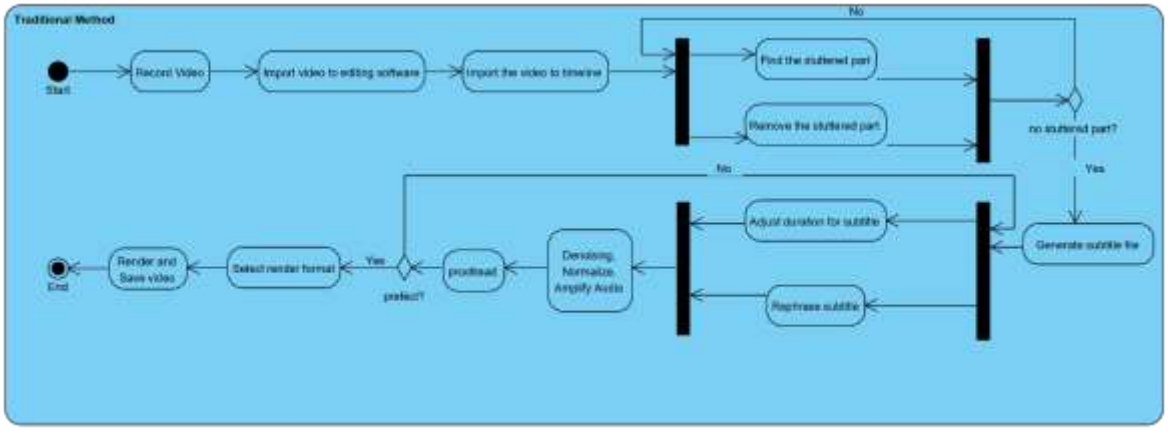

Figure 3.11 Activity Diagram - Traditional Method

Overall system flow

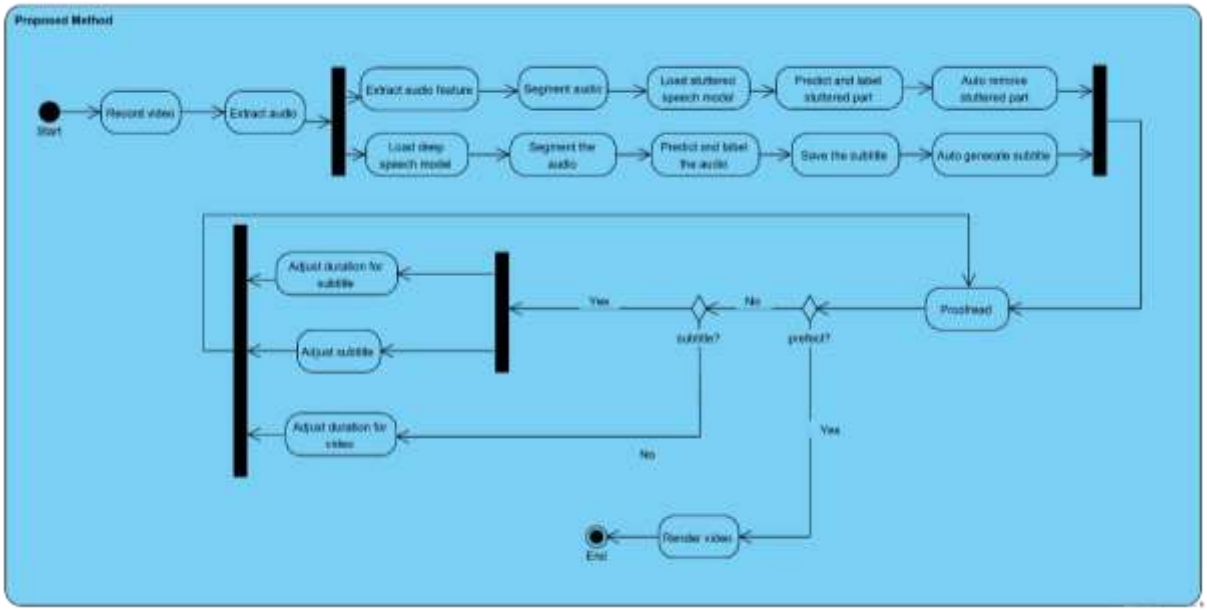

Figure 3.12 Activity Diagram - Proposed System

Record Video

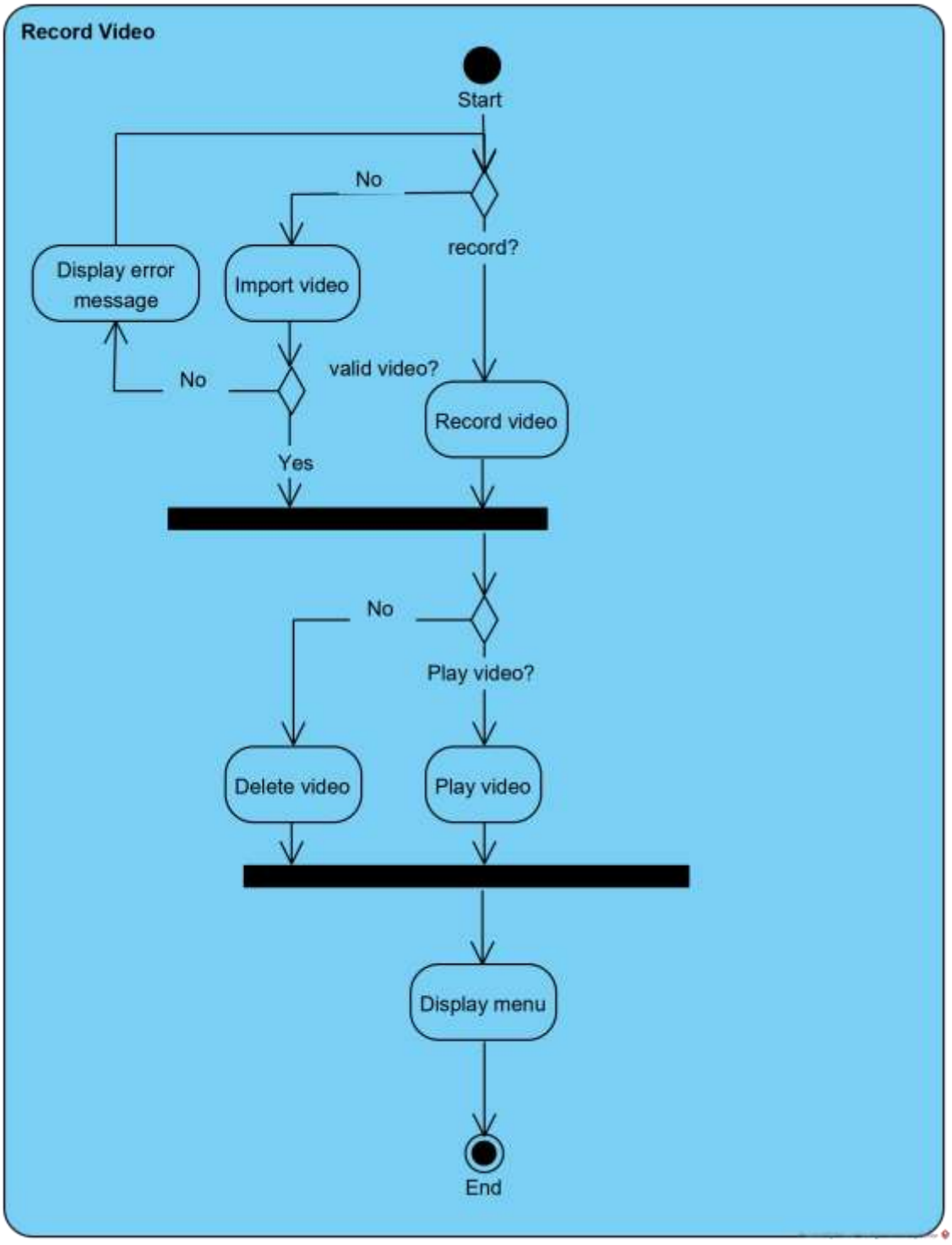

Figure 3.13 Activity Diagram - Record Video

Edit Video

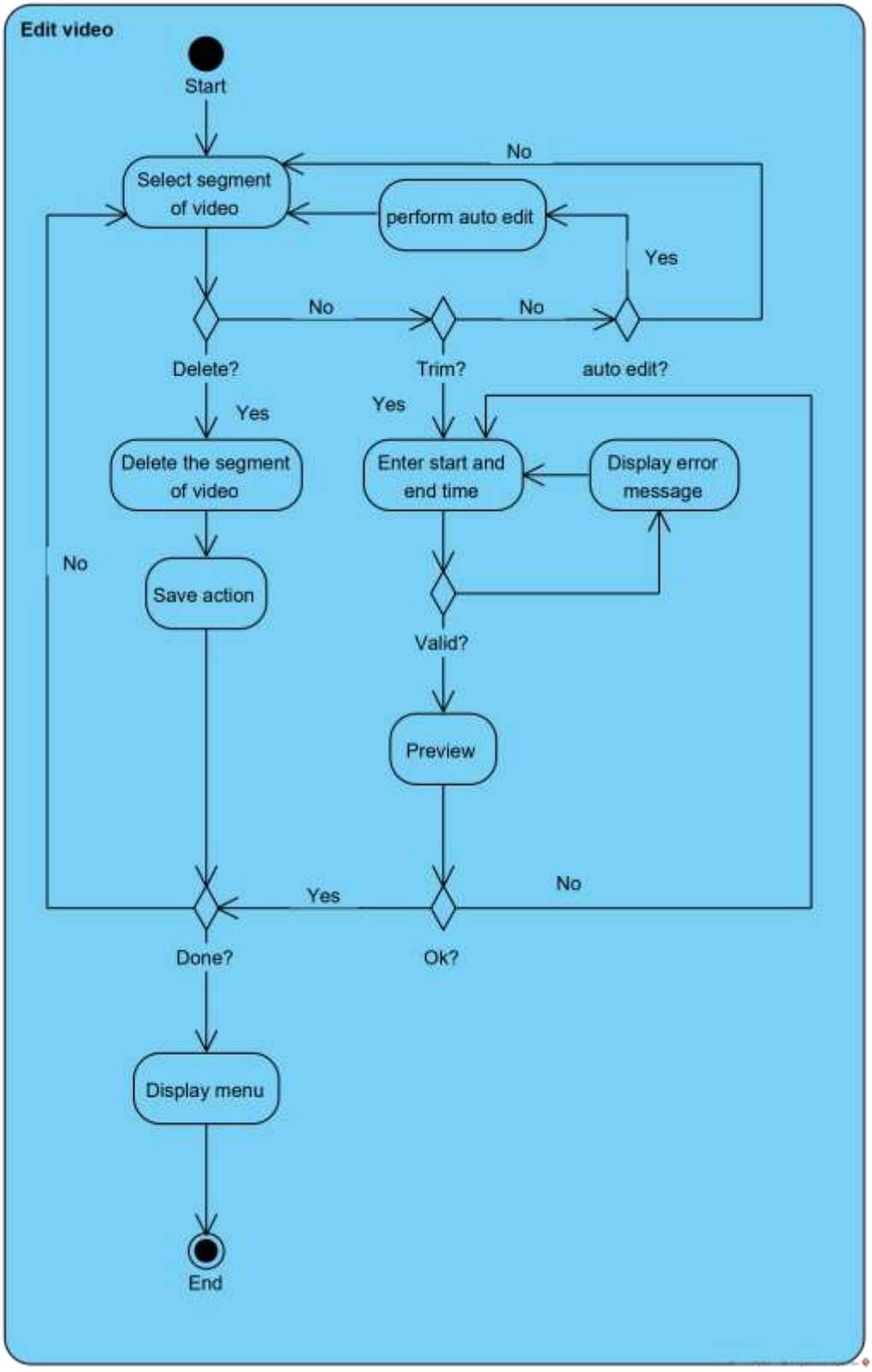

Figure 3.14 Activity Diagram - Edit Video

Edit Subtitle

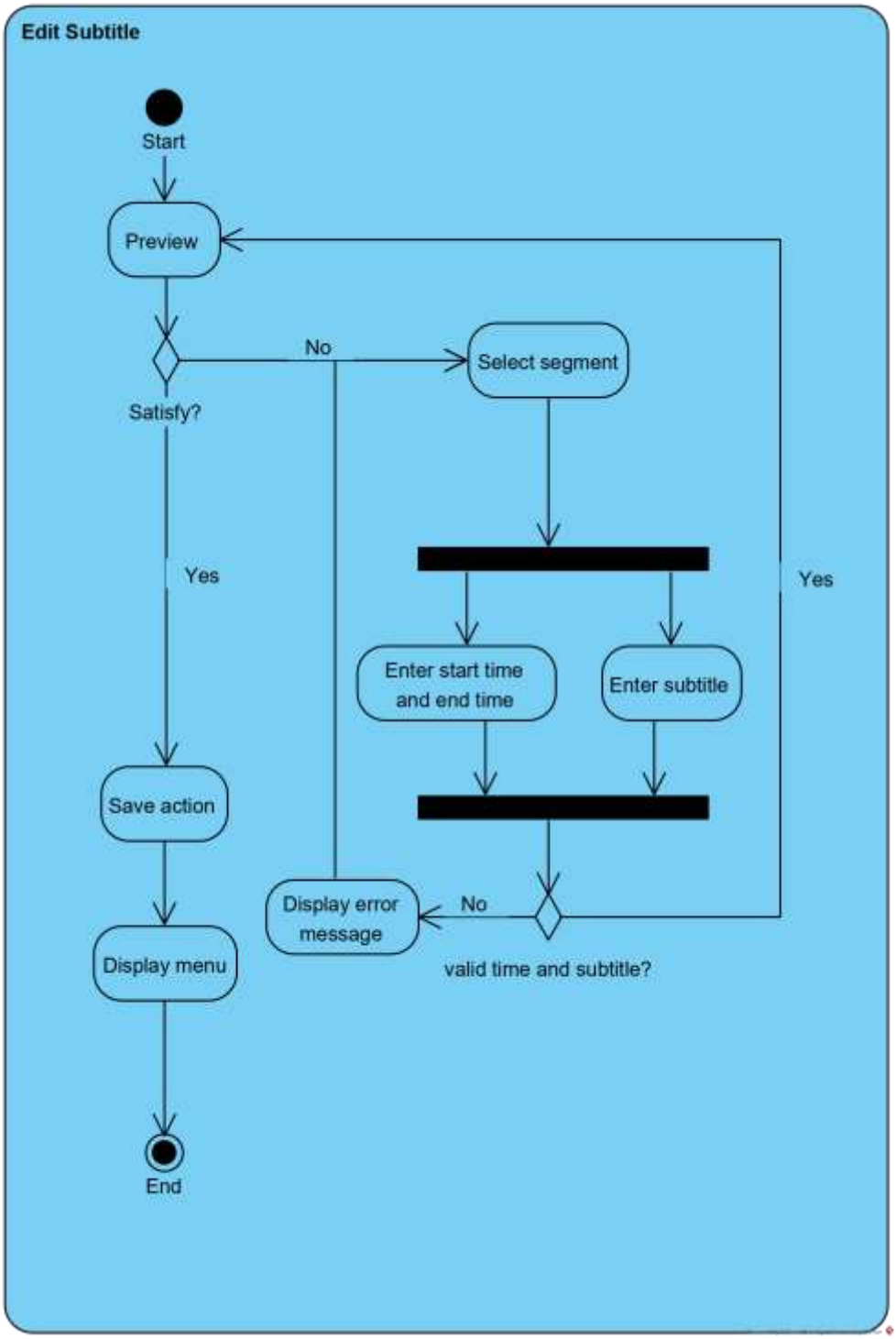

Figure 3.15 Activity Diagram - Edit Subtitle

3.5.2.6 Class Diagram

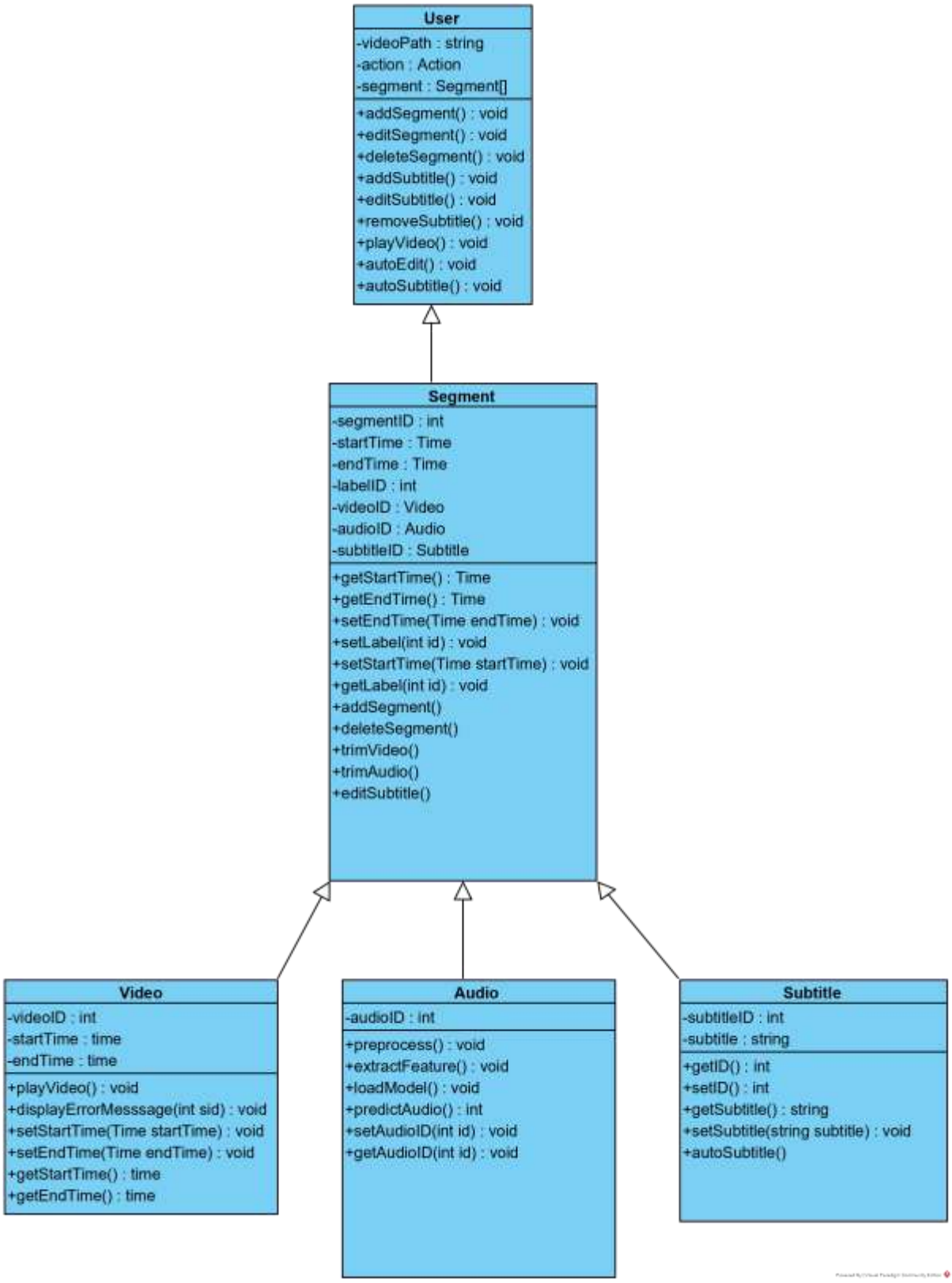

Figure 3.16 Class Diagram

3.5.2.7 Sequence Diagram

Record Video

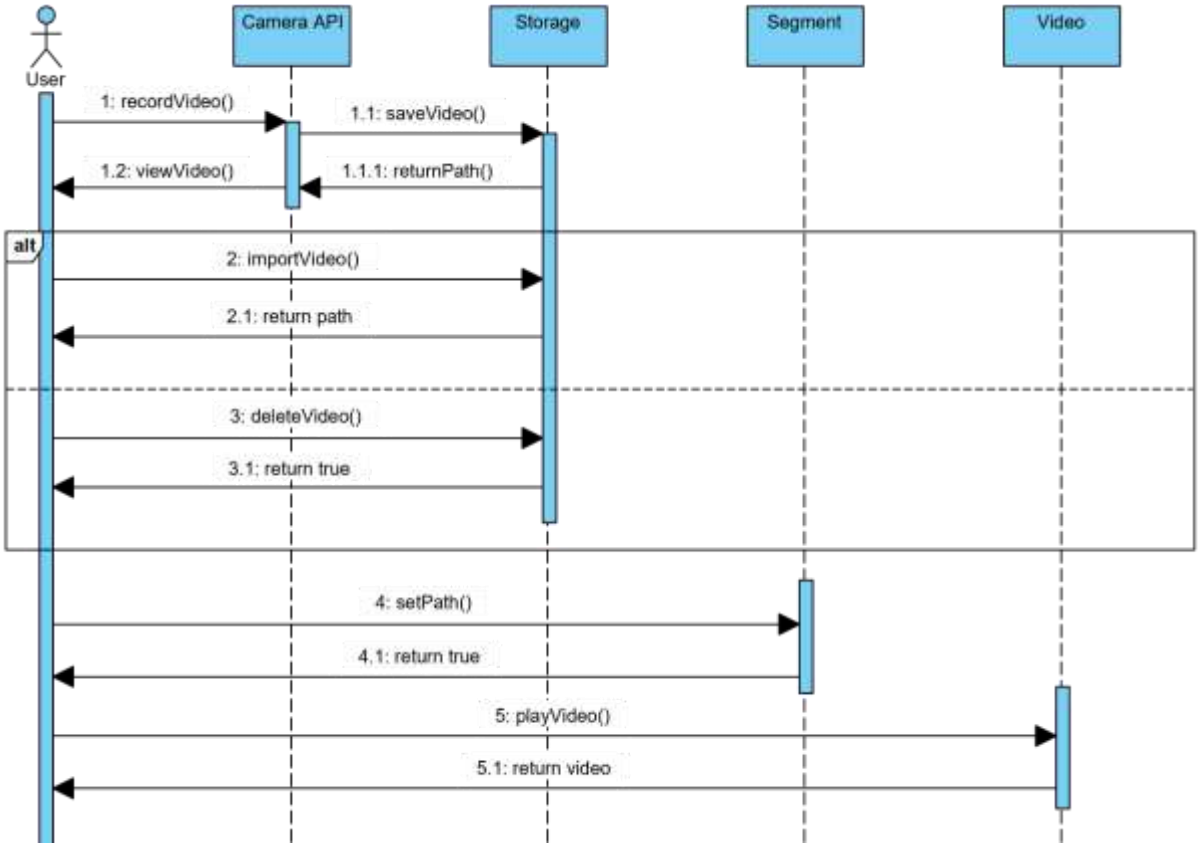

Figure 3.17 Sequence Diagram - Record Video

Edit Video

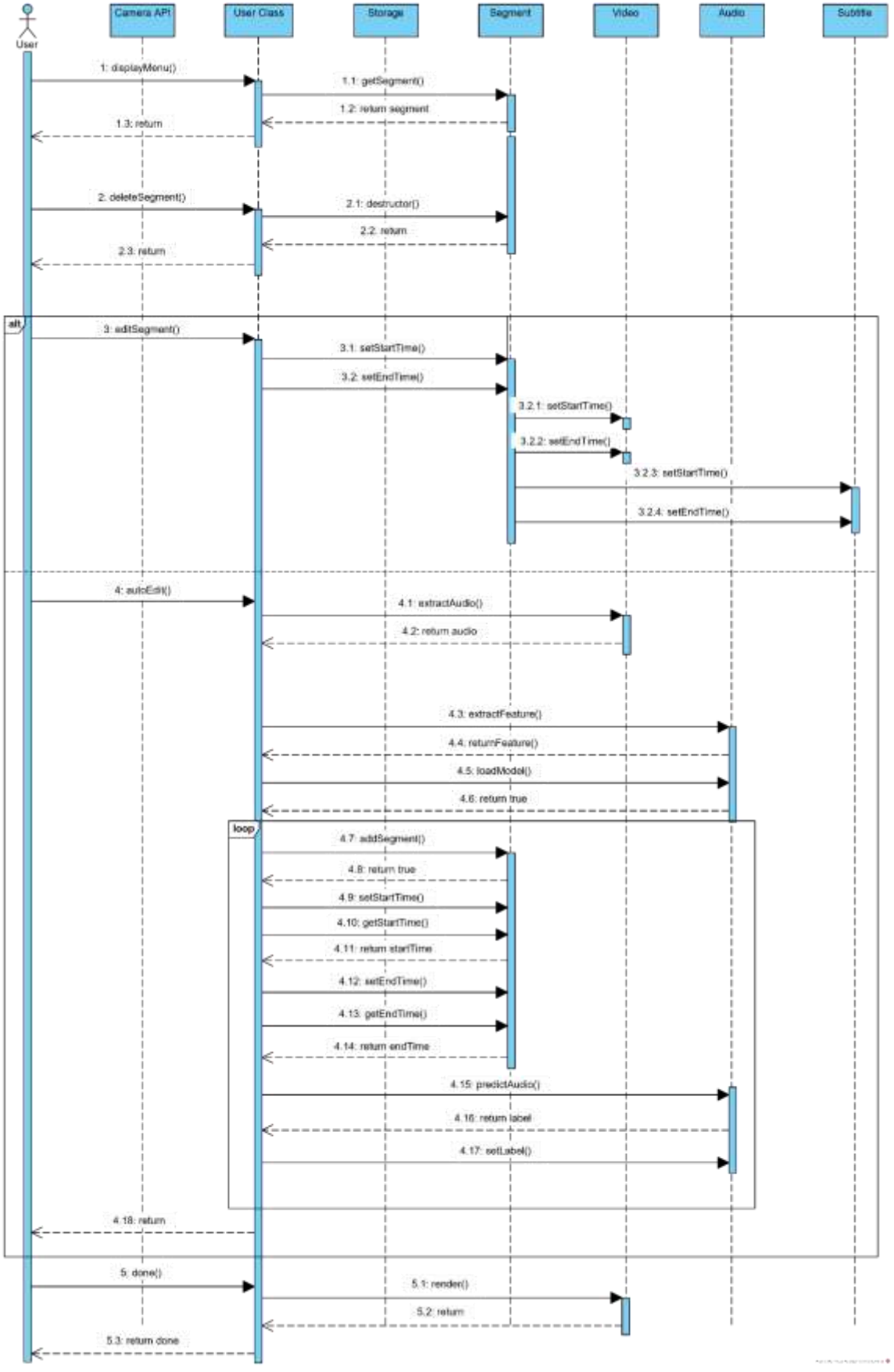

Figure 3.18 Sequence Diagram - Edit Video

Edit Subtitle

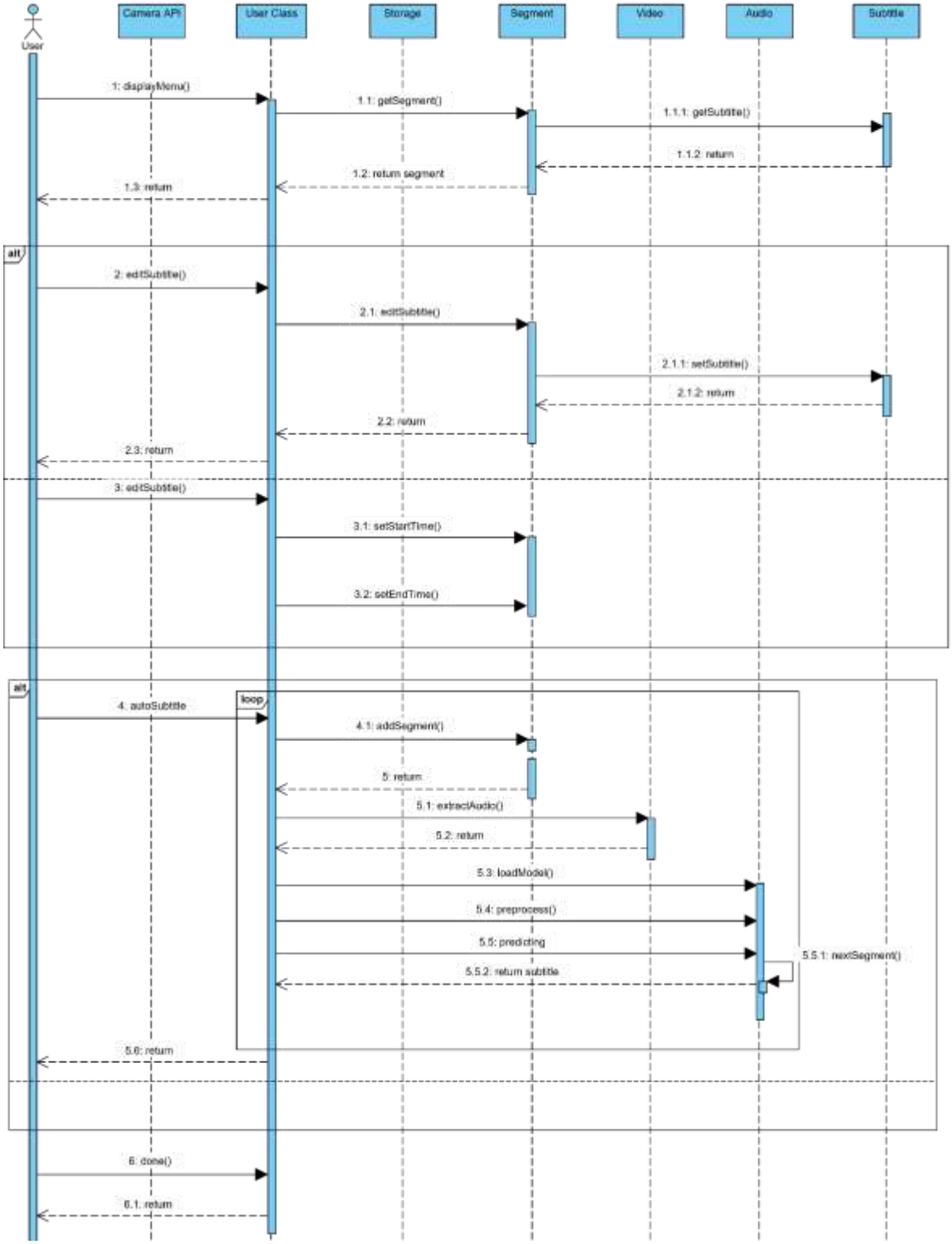

Figure 3.19 Sequence Diagram - Edit Subtitle

3.5.2.8 State Diagram

Record video

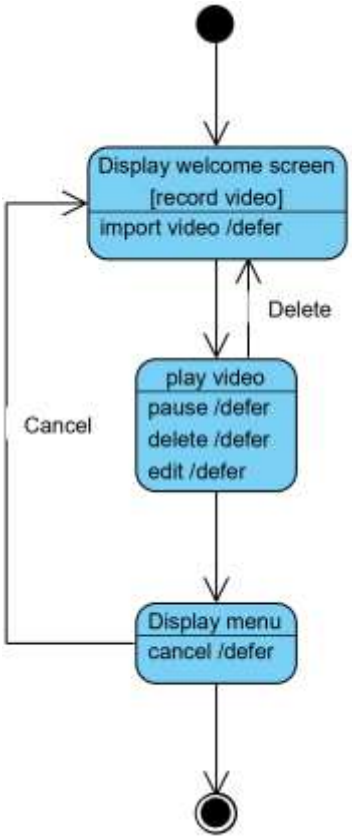

Figure 3.20 State Diagram - Record Video

Edit Video

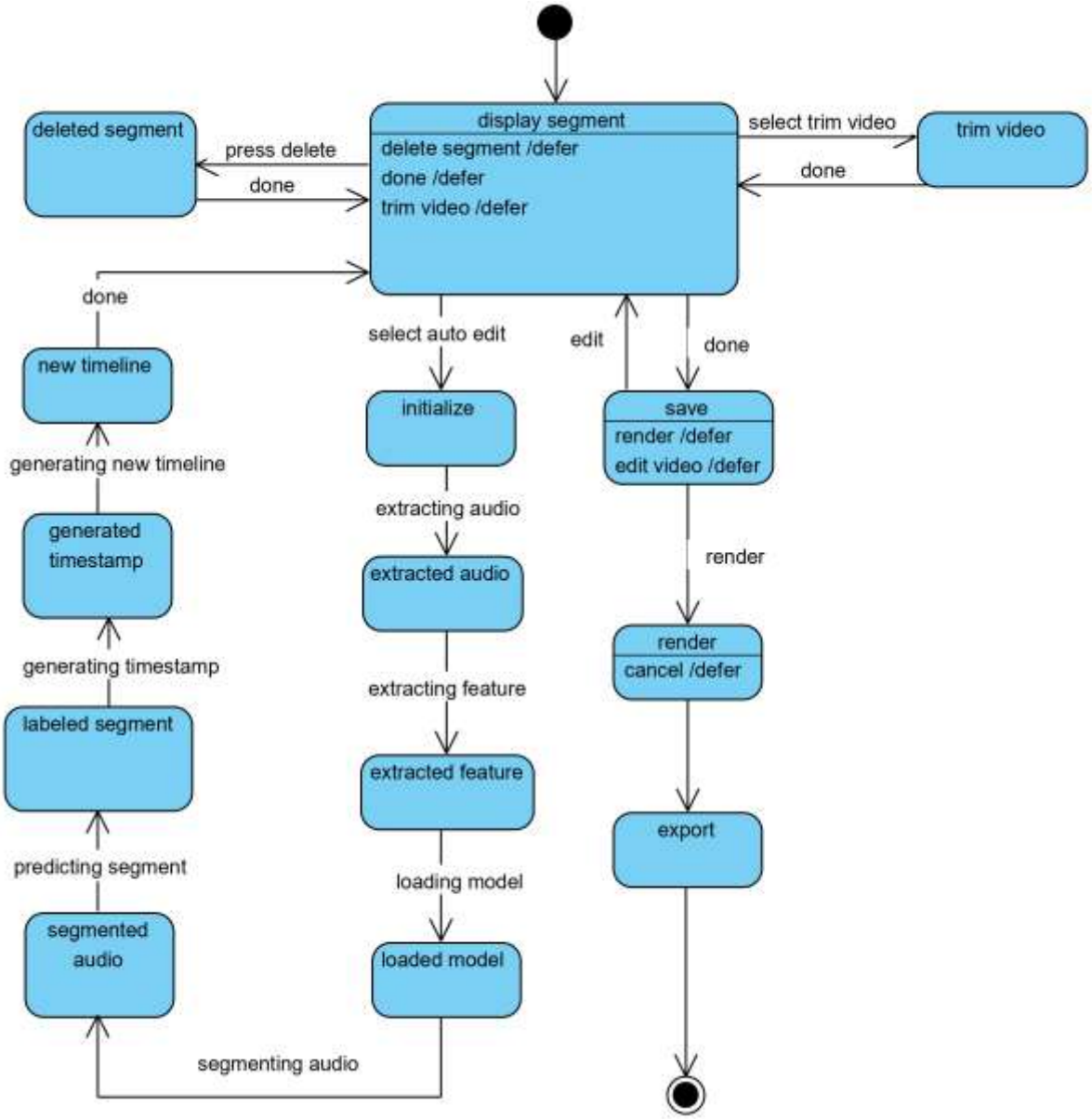

Figure 3.21 State Diagram - Edit Video

Edit Subtitle

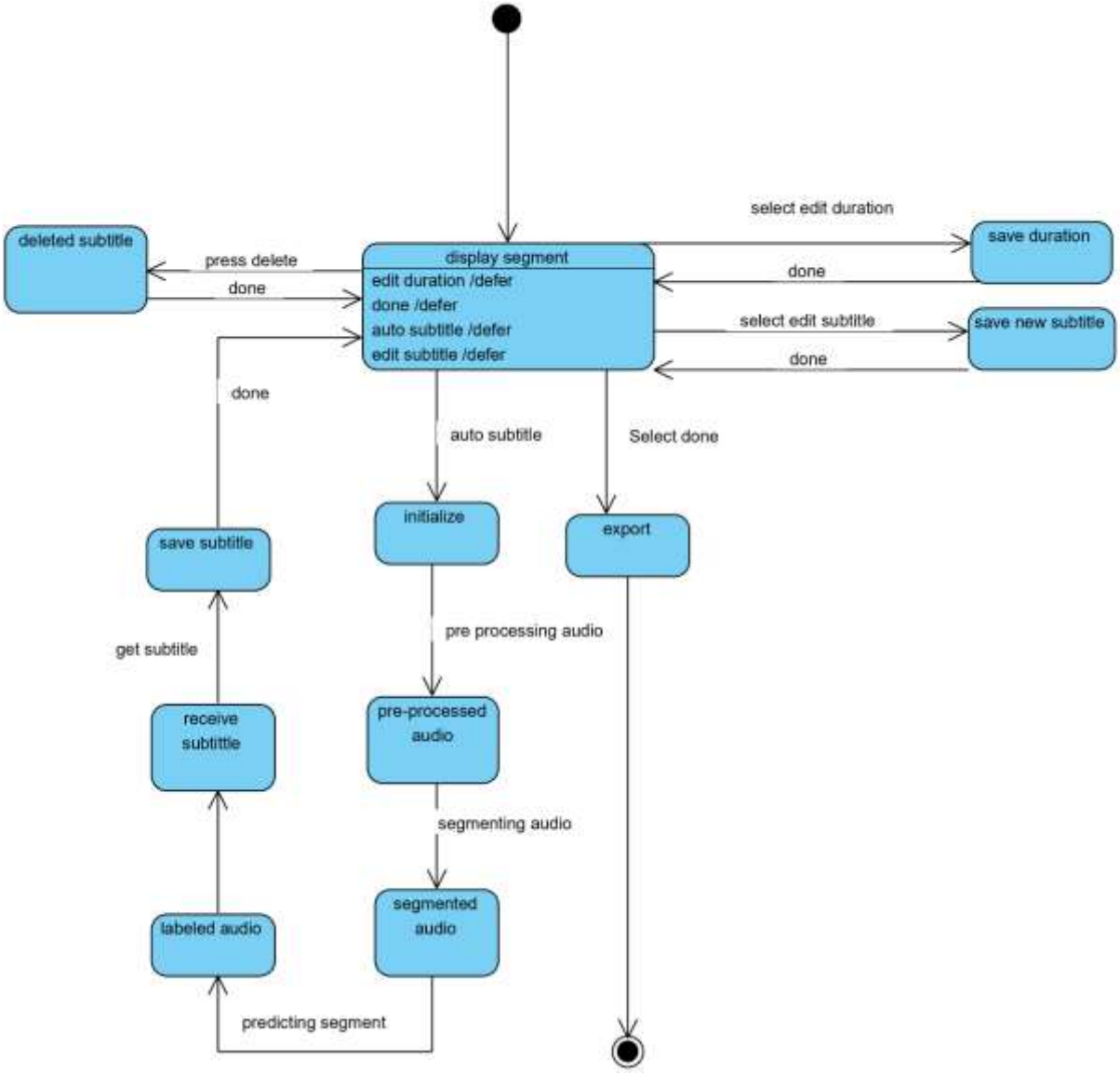

Figure 3.22 State Diagram - Edit Subtitle

3.5.2.9 Package Diagram

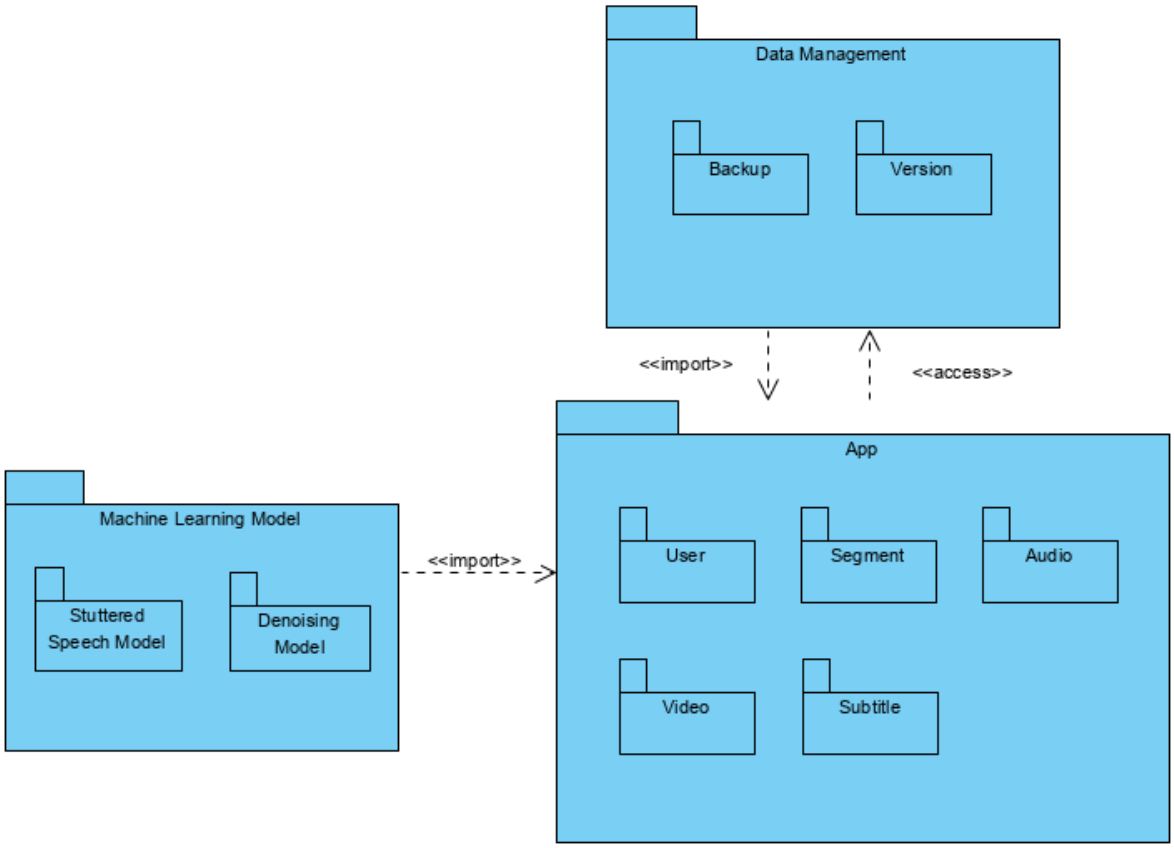

Figure 3.23 Package Diagram

3.5.2.10 Deployment Diagram

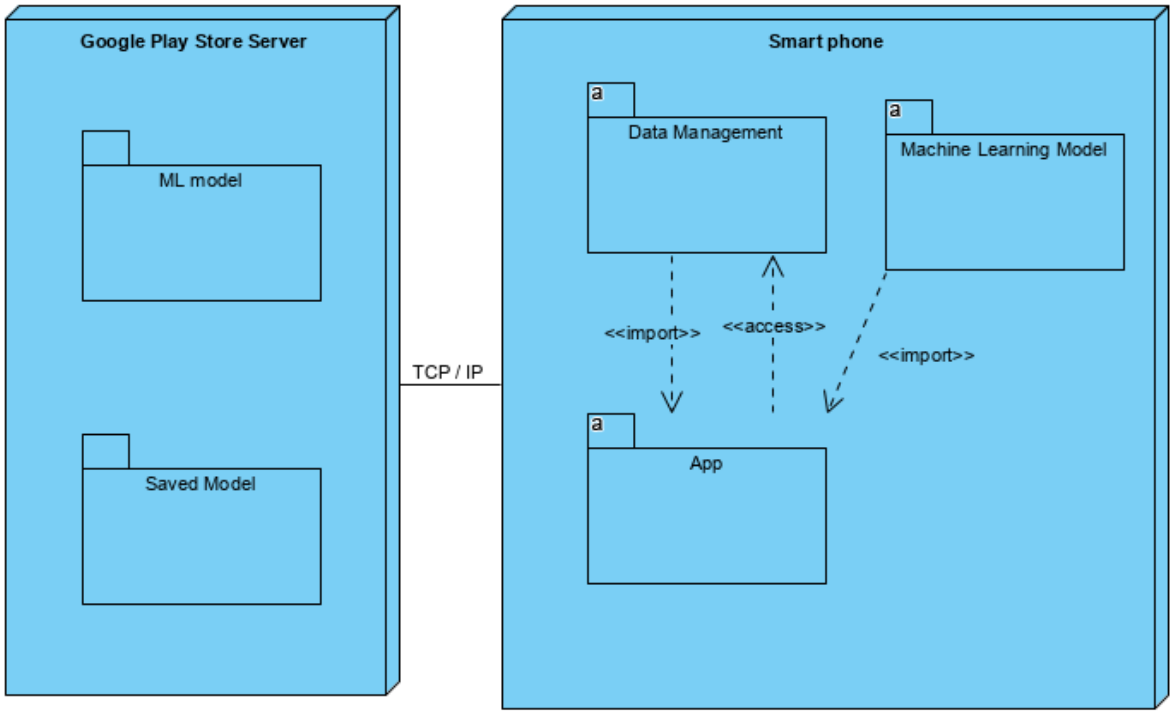

Figure 3.24 Deployment Diagram

3.5.3 User Interface

Home

Figure 3.25 UI - Home Screen

Import

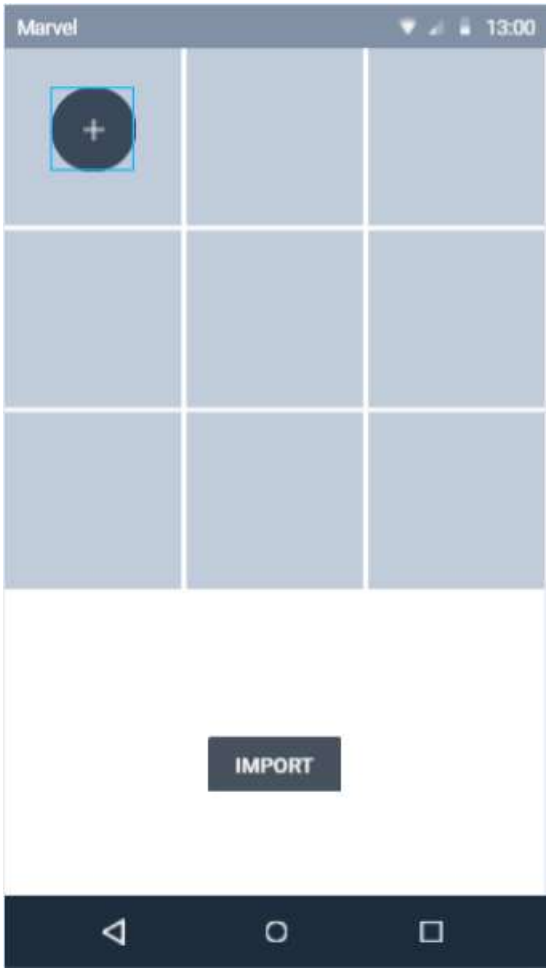

Figure 3.26 UI - Import Screen

Edit Video

Figure 3.27 UI - Edit Video Screen

Edit Subtitle

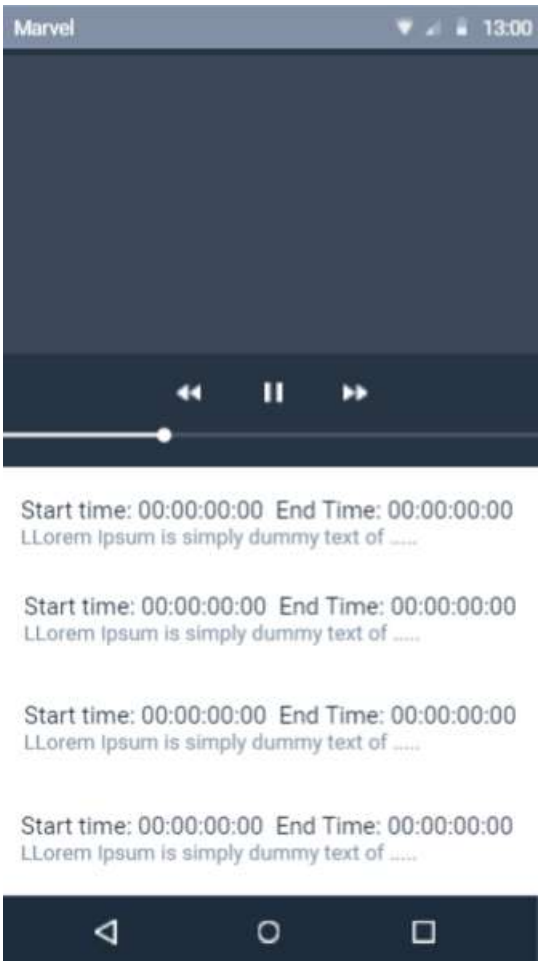

Figure 3.28 UI - Edit Subtitle Screen

4 IMPLEMENTATION

4.1 Testing

4.2 Output Analysis

5 CONCLUSION

5.1 Finding & limitations

5.2 Contribution

5.3 Summary

5.4 Future Enhancement

REFERENCES

Celie O'Neil-Hart, Howard Blumenstein. (2016, April). *The latest video trends: Where your audience is watching*. Retrieved from Think With Google: https://www.thinkwithgoogle.com/consumer-insights/video-trends-where-audience-watching/

Fernandes, E. A. (2015, Jan 21). *Scrum explained*. Retrieved May 2020, from Code Project: https://www.codeproject.com/Articles/704720/Scrum-explained

MathWorks. (n.d.). *Unsupervised Learning*. Retrieved from MathWorks: https://www.mathworks.com/discovery/unsupervised-learning.html

Mobile & Tablet Android Version Market Share Worldwide. (2020). Retrieved 7 5, 2020, from statcounter: https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide

scrum.org. (n.d.). *What is a Product Backlog?* Retrieved May 2020, from Scrum.org: https://www.scrum.org/resources/what-is-a-product-backlog

Scrum.org. (n.d.). *What is a Product Owner?* Retrieved from Scrum.org: https://www.scrum.org/resources/what-is-a-product-owner

Stecanella, B. (2017, June 22). *An Introduction to Support Vector Machines (SVM)*. Retrieved from MonkeyLearn: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

stutter. (n.d.). Retrieved from Cambridge Dictionary: https://dictionary.cambridge.org/dictionary/english/stutter

Swapnil D. Waghmare, Ratnadeep R. Deshmukh, Pukhraj P. Shrishrimal, Vishal B. Waghmare, Ganesh B. Janvale and Babasaheb Sonawane. (2017, June). A Comparative Study of Recognition Technique Used for Development of Automatic Stuttered Speech Dysfluency Recognition System. *Indian Journal of Science and Technology*, 2-13.

Vikhyath Narayan K N, S P Meharunnisa. (2016). Detection and Analysis of Stuttered Speech. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), 5*(4), 953-955.

Youtube for Press. (n.d.). Retrieved from Yotube: https://www.youtube.com/about/press/

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., … Zhu, Z. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. *33rd International Conference on Machine Learning, ICML 2016*, *1*, 312–321.

Dash, A., Subramani, N., Manjunath, T., Yaragarala, V., & Tripathi, S. (2018). Speech Recognition and Correction of a Stuttered Speech. *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*, 1757–1760. https://doi.org/10.1109/ICACCI.2018.8554455

Mecs, M. (2016). *Detection and Analysis of Stuttered Speech*. *5*(4), 952–955.

Waghmare, S. D., Deshmukh, R. R., Shrishrimal, P. P., Waghmare, V. B., Janvale, G. B., & Sonawane, B. (2017). A Comparative Study of Recognition Technique Used for Development of Automatic Stuttered Speech Dysfluency Recognition System. *Indian Journal of Science and Technology*, *10*(21), 1–10. https://doi.org/10.17485/ijst/2017/v10i21/106092

APPENDICES

I. Appendix A – WBS
II. Appendix B – Gantt Chart
III. Appendix C – Network Diagram